Escuela Politécnica Superior

Simulación de una moneda virtual con Blockchain.

Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:
Juan Serna Jaén

Tutor/es:
Santiago Meliá

Universitat d'Alacant
Universidad de Alicante

Abril 2017

Prefacio

Investigando en referencia a cuál podría ser mi proyecto de final de carrera, quería buscar un objetivo en el que pudiera poder a prueba todo (o en gran parte) lo aprendido en la Universidad de Alicante. Ya sea desde los primeros análisis, hasta la creación de un desarrollo de programación haciendo ver un concepto o avance nuevo que pudiera dar una explicación a alguna tecnología novedosa.

Otro punto a tener en cuenta durante esta búsqueda fue que la investigación debía estar basada en código abierto, para que pudiera seguir siendo de utilidad a la Universidad de Alicante tras el fin de mi proyecto.

Por lo que mi investigación me llevó al aprendizaje sobre la creación de las diferentes monedas virtuales que estaban apareciendo en el mercado, de carácter público y basadas bajo la misma metodología que había comenzado la moneda Bitcoin.

En un principio mi objetivo era crear una simulación de una moneda virtual y dar a conocer a la universidad cómo funciona, pero conforme iba aprendiendo más acerca de cómo se creaban las monedas virtuales encontré lo que realmente hacía funcionar no solo a la creación de las monedas virtuales, sino a cualquier tipo de contrato, programa, desarrollo, … bajo una red descentralizada llamada Blockchain.

Por lo que, como resultado; mi proyecto se divide en el análisis de una moneda virtual, la investigación de una cadena de bloques (y el potencial que se le puede dar en el mundo real) y, finalmente; una simulación entre ambos conceptos gracias a un desarrollo dividido en tres partes.

Agradecimientos

De forma especial a mi tutor Santiago Meliá, al que agradezco enormemente la disposición y la ayuda que me ha ofrecido al realizar este proyecto. Gracias a él ha sido posible crear este proyecto.

También a todo el profesorado de Ingeniería Informática de la Universidad de Alicante, dado que soy quién soy gracias a todo su conocimiento.

Finalmente, y no menos importante; a mi familia que siempre ha estado apoyándome desde el primer momento hasta el final, en especial a esa persona que siempre he tenido a mi lado y me ha ayudado en todo momento.

Índice de contenidos

1. Justificación y Objetivos

El motivo por el que expongo mi trabajo y lo enfoco a la "simulación de una moneda virtual" viene dado por tres temas principalmente.

Para el proyecto final de carrera quería hacer un trabajo en el que aprovechara el máximo posible todo lo aprendido durante la universidad, por lo que investigando me di cuenta que el campo de las criptomonedas se apoyaba en un conjunto de metodologías software basados (entre otros muchos) en los siguientes campos:

- Criptografía: la más importante de todas, el núcleo de toda actividad de cualquier moneda virtual reside en lo que se conoce como "blockchain" y todo el tráfico de información requiere un importante conocimiento de cifrado. El más común es el SHA-256.
- Modelo distribuido (P2P): uno de los conceptos teóricos más importantes en el mundo de las monedas virtuales es la descentralización. Es decir, la moneda ya no depende de una "entidad" que la avalara para proporcionarle valor y todo esto se consigue mediante un sistema en el que todos los ordenadores que componen la red de la criptomoneda, tienen la información de todos, lo que la hace también más segura y fiable (este es el segundo principio, como veremos más adelante; de cómo funciona blockchain).
- Servicios Web / API: para poder conectar el núcleo (blockchain) a las diversas plataformas que lo utilizan (mineros, usuarios, webs de transacciones, aplicaciones de compra y venta, …) se requiere un amplio soporte a la hora de ofrecer servicios para que otras entidades puedan conectarse para poder recibir la información que necesiten.
- Bases de Datos: para guardar toda la información tanto del núcleo como de los propios usuarios.
- Aplicaciones de escritorio / web: para ofrecer a los usuarios todas las facilidades para poder utilizar su criptomoneda.

El segundo es la consolidación de una figura abstracta, (en nuestro caso una moneda virtual) con un valor en los mercados internacionales e independiente de emisores centrales y gobiernos. Un ejemplo ocurre con la moneda virtual Bitcoin, que poco a poco ha ido ganando mucho peso con el paso de los años (hasta el punto de que a día de hoy 1 Bitcoin equivale a 1200 dólares) ya que ha ido contando con el soporte de grandes empresas a nivel internacional. Otro ejemplo que, curiosamente está siendo respaldado (y cada vez más) por los bancos a nivel mundial, es la moneda virtual conocida como Ether. La importancia de esta moneda no radica en su propio valor, sino en la red de blockchain que tiene conocida como Ethereum; por lo que esto me lleva a la justificación número 3 y la que considero que es más importante.

Cuando investigaba acerca de cómo funcionan las monedas virtuales, me percaté de que realmente lo más importante no era el hecho de crear una moneda, sino la red en la que se distribuía de forma descentralizada, y por ello mi última justificación es **Blockchain**.

La descripción más exacta de la cadena de bloques conocida como Blockchain sería la de "un registro distribuido resistente a la sincronización y sin necesidad de confianza entre los miembros que la conforman". Y es que, gracias precisamente a esa descentralización, estamos ante uno de los sistemas más seguros de internet dado que todos los ordenadores de la red

tienen la misma información y para que pueda haber un cambio de esta información todos los ordenadores de la red deben de confirmarlo.

Los objetivos de este proyecto serán dar respuesta a las siguientes preguntas:

- Referente a una moneda virtual
    - ¿Qué es una moneda virtual?
    - ¿Para qué sirve?
    - ¿Cómo funciona?
    - ¿Cómo se intercambia?
- Referente a una cadena de bloques (Blockchain)
    - ¿Qué es blockchain?
    - ¿Para qué sirve?
    - ¿Cómo funciona?
    - Utilidades en el ámbito real / empresarial.

Y además ofrecer un ejemplo práctico de cómo funciona una moneda virtual con las siguientes características:

- Blockchain: para poder recrear las deferentes transacciones de nuestra moneda virtual.
- Minero: para poder verificar estas transacciones.
- Web de intercambio de moneda: para poder ver cómo funciona un sitio de transacciones.

2. Crypto-currency

2.1. Introducción

El intercambio de monedas virtuales está presente en la actualidad de manera muy activa, desconocido por muchos, está siendo uno de los medios de transacción más revolucionarios de todos los tiempos dado que a diferencia de otras monedas, es una divisa electrónica libre de bancos y estados, pero por lo general también es uno de los métodos más desconocidos.

Este apartado del proyecto tiene como fin responder a todas las preguntas acerca de su funcionamiento y comprobar si este sistema tan novedoso es útil, seguro y si puede constituir una manera alternativa de pago respecto al resto.

Junto con esta memoria, crearemos también nuestra propia moneda virtual “simulada” y veremos cómo funciona a diferentes niveles.

2.2. Historia de Bitcoin

Puesto que es la primera moneda virtual conocida y (salvando las diferencias) es el modelo a seguir de las otras monedas virtuales (incluida la de este proyecto) hablaremos sobre cómo se originó Bitcoin y qué es.

El concepto de una moneda virtual que fuera descentralizada y utilizara la criptografía surgió por primera vez por Wei Dai en 1998.

Con la crisis financiera en Estados Unidos en el año 2008, genera una recesión a nivel global (debido a una importante inyección de liquidez en los bancos) que hace que la gente empiece a desconfiar de las divisas gestionadas de una forma centralizada y empiece a buscar divisas alternativas.

Desde que existe internet, ya se han intentado (sin éxito) diferentes métodos para poder crear una moneda virtual pero el principal problema radicaba en la confianza. Dado que una moneda virtual es un concepto abstracto (información) solo se podía mover en dos tokens diferentes, si la moneda se había gastado o no. Para ello el problema residía en que había que tener una fuente centralizada de información, con lo que al haber solo un punto de ruptura el sistema era muy vunerable.

Por lo que, en 2008; Satoshi Nakamoto publicó la primera prueba de concepto del protocolo Bitcoin de forma abierta, planteando un sistema de bloques distribuido llamado Blockchain con lo que todas las transacciones quedaban anotadas y eran verificadas de forma contínua por la totalidad de los ordenadores que había en la red. De esta forma se solventaba el problema de la vulnerabilidad debido a que para que una transacción sea correcta, todos los ordenadores de la red deben de tener la misma información.

En 2009 nace la red Bitcoin publicando la primera cartera (cliente) con la que los usuarios podían guardar los bitcoins.

A partir de 2010 Satoshi deja a un lado el desarrollo del proyecto dando paso a cientos desarrolladores (gracias al código abierto del propio proyecto) y otras tantas monedas virtuales que seguirán la misma base que Bitcoin.

Desde ese año hasta la actualidad, Bitcoin es una moneda que se ha ido fortaleciendo y consolidando cada vez más, hasta tal punto de que actualmente su valor es de 1 BTC = 1.200 $ (los primeros cambios públicos con esta divisa tenían un valor de 1 BTC = 0,003$).

2.3. Para qué sirve

Bitcoin, al igual que cualquier otra moneda, sirve para ofrecer bienes y servicios a cambio de un pago. El valor de la moneda viene dado por los mercados o también cómo de fuerte esté en ese momento. Actualmente el valor de la moneda está en torno a unos 1.200 dólares y cada vez son más las empresas que permiten el pago con la moneda virtual.

A continuación, mostramos las ventajas y desventajas que (tanto Bitcoin como cualquier otra moneda virtual) pueden tener.

Ventajas

- Bitcoin tiene un límite de emisión de 21 millones, por lo que no se devaluaría.
- Es una moneda basada en una red descentralizada, por lo que no es controlada por ningún ente (es decir, es una moneda anónima)
- El control de las transacciones se realiza a través de todos los participantes de la red, por lo que la moneda es segura.
- Transacciones en tiempo real (cada transacción puede durar entre 15 y 20 minutos, dependiendo de la complejidad del minado)

Desventajas

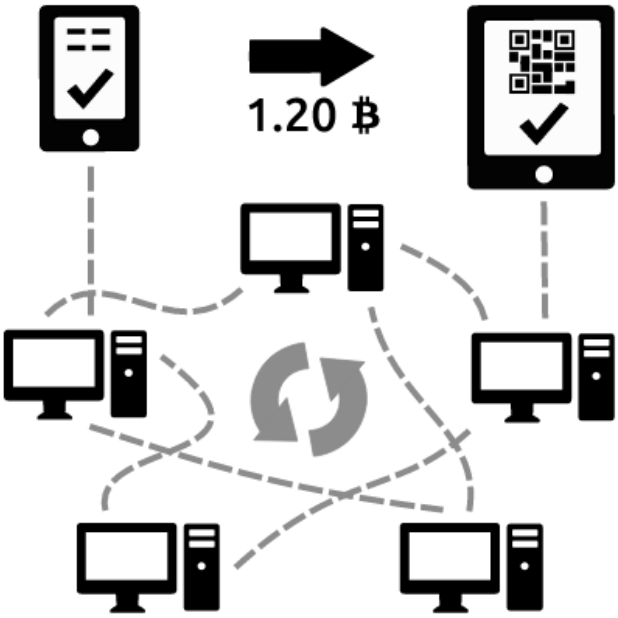
- La moneda virtual (al no estar regulada por ningún organismo) depende exclusivamente de la oferta y la demanda.
- No hay garantías de que se convierta en una moneda aceptada por todos.

2.4. Cómo funciona

En términos funcionales, un usuario simplemente necesita tener una cartera de bitcoins (que puede estar en su propio ordenador o en una web especializada en crear carteras) y ya está. A nivel de usuario funciona igual que una cuenta bancaria o un correo electrónico, tenemos una dirección que será nuestro monedero de bitcoins que podremos utilizar para enviar dinero o viceversa.

Para que el sistema de bitcoins lleve a cabo todas las cuentas de transacciones, existe una red pública que se conoce como Blockchain. Es un sistema de contabilidad (encriptado) distribuido y público en el que se van añadiendo bloques. Cada bloque se compone de un conjunto de transacciones verificadas (o confirmadas), uniéndolo al resto de bloques mediante un valor Hash que ha sido previamente descifrado por la comunidad de mineros de Bitcoin. Una vez se añade la cadena de bloques nueva, para poder realizar otra transferencia el usuario está obligado a actualizarse la cadena de valores para que toda la red esté sincronizada con los mismos datos.

Este sistema hace que Bitcoin sea imposible de falsear puesto que para intentar añadir un bloque "falso" debemos de falsear toda la cadena de bloques existente, con sus correspondientes hashes y además falsear también todos los ordenadores de la red para que puedan validad que ese bloque es realmente correcto.



Cada una de estas transacciones se componen de una clave pública y una clave privada. La clave pública, al igual que la dirección la puede ver cualquier usuario que tenga acceso a la red de Blockchain (de esta forma se queda guardado todo un registro público de transacciones)

mientras que la clave privada es utilizada para firmar las transacciones de forma que esta no se vea alterada.

El paso final de las transacciones es incluirlas en un bloque que se añadirá a la cadena de blockchain, para ello existe un proceso denominado minería, en el cual un conjunto de ordenadores (de manera distribuida) que confirman que la transacción ha sido correcta. Cuantas más confirmaciones se hayan realizado por diferentes ordenadores, más segura será la transacción. Estas transacciones se empaquetan en un bloque ajustándose a normas de cifrado que veremos en el siguiente punto, de forma que impide que cualquier bloque se modifique. Una vez codificado se añade a la cadena y se vuelve a empezar el ciclo.

2.5. Criptografía

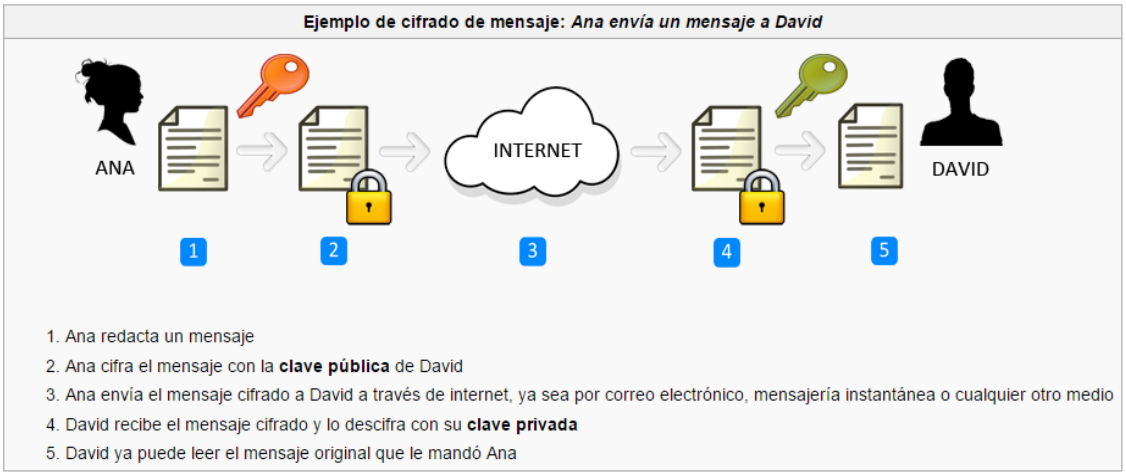
Desde la época de 1970, la utilización de firmas digitales basadas en criptografía de clave pública (criptografía asimétrica) ha ido en aumento gracias a su seguridad.

La Criptografía Asimétrica consiste en un par claves que se utilizan en los envíos de mensajes para reforzar la seguridad de estos de forma que si no se tiene acceso a estas claves sea imposible descifrarlo.

Por una parte, tenemos la clave pública; la cual es accesible por todos y es la que debemos de pasar a la otra parte implicada para que nos pueda mandar un mensaje.

Por otra, tenemos la clave privada; solo la sabe el destinatario del mensaje y sirve para poder desencriptar la información que se nos ha enviado.

De esta forma todo el mundo puede enviarnos mensajes (gracias a la clave pública) mientras que solo nosotros podremos abrirlo (clave privada).

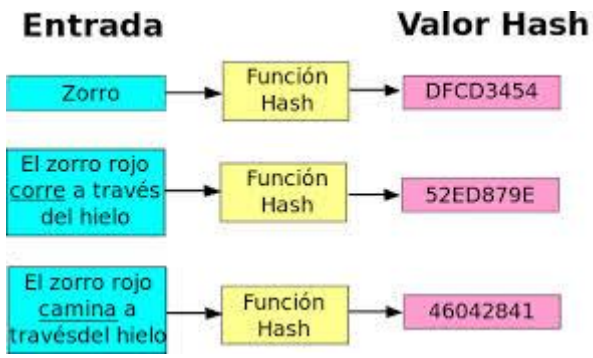


Para poder generar las claves privadas y públicas tenemos otra función criptográfica denominada Hash.

Hash (resumen criptográfico) es un algoritmo que convierte una cantidad de datos grande en un número (por lo general en hexadecimal) de tamaño fijo. Es utilizado para generar las claves públicas a partir de claves privadas dado que a partir de una clave privada es muy fácil generar el código hash, pero al revés es casi imposible.

De esta forma en una moneda virtual (en concreto Bitcoin) dada nuestra clave privada podremos crear nuestra clave pública y nuestra dirección pública.

El hash también sirve a la hora de crear bloques nuevos en nuestra cadena de bloques (blockchain) ya que estos están conectados entre ellos por los valores hash (que previamente han tenido que minar).



Para la generación del valor hash Bitcoin utiliza el algoritmo criptográfico SHA-256 (Secure Hash Algorithm) de forma que genera un hash de 64 dígitos hexadecimales de tamaño fijo de 256 bits (32 bytes).

De forma que la frase "Esto es un ejemplo" crea el siguiente valor hash con SHA-256 "38e1cad08cd88efd203280451c0a415454a5d2c14c1a0d79bc5c77d295726cc5".

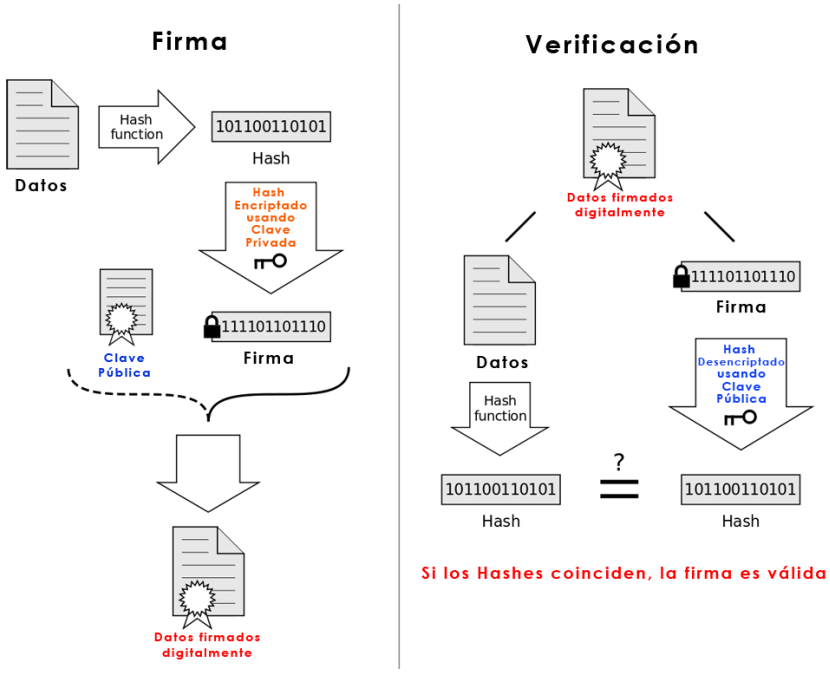
Si quisiéramos pasar el hash obtenido a la frase "Esto es un ejemplo" tendríamos que intentar obtenerlo por un ataque de fuerza bruta en el que serían necesarias una inmensa cantidad de operaciones (para el algoritmo SHA-1, que es su predecesor el número total de operaciones necesarias está al orden de 2 elevado a 63 (2^{63})).

Cada uno de estos hashes dentro de la moneda virtual debe de ser único por lo que no hay dos hashes iguales. En el caso de que sucediera esto estaríamos hablando de una "colisión". Para hacernos una idea, SHA-1 (de 160 bits) fue lanzado en 1995 y ha sido este año (2017) que Google ha encontrado su primera colisión (creación de dos hashes idénticos).

2.6. Transacción de Bitcoins

Suponiendo que dos personas cuentan con un monedero de Bitcoins, vamos a ver qué pasa realmente cuando se realiza una transacción de Bitcoins.

- Persona A genera una transacción de 3 Bitcoins a Persona B, esta transacción se compone de una Clave Pública y una Clave Privada (Firma) de la Persona A.
- Se genera un código Hash y se firma con la clave privada, se le añade la clave pública para comprobar la firma y el monedero de Persona A suelta la transacción a la red de Bitcoin para que sea procesada.



- Persona B recibe la transacción, efectua un Hash del mensaje y desencripta la firma con la clave pública, compara el Hash del mensaje con el de la firma para ver que coinciden.
- La transacción se envía a los nodos de Blockchain.
- Estos nodos comprueban la transacción y los envían a sus nodos sucesivos.
- La transacción llega a unos nodos que están minando, validan de nuevo la transacción y además la incluye en un bloque (conjunto de transacciones ya validadas y agrupadas).
- Los mineros cogen el hash del bloque anterior y añaden el conjunto de transacciones que ellos tienen en su bloque, añaden un número (Nonce) y calculan el Hash de nuevo.
- Si el hash comienza por un determinado número de ceros (Dificultad) habrán encontrado el bloque.
- Si no, generarán un nuevo Hash incrementando el Nonce.
- Una vez confirmado el bloque, se actualiza toda la cadena añadiéndolo con el hash creado y el minero recibe una cierta cantidad de Bitcoins.

3. BlockChain

3.1. Introducción

Lo que hace que una crypto-currency funcione perfectamente y con las características que hemos descrito es gracias al núcleo central de este tipo de divisas, el Blockchain (cadena de bloques).

La descripción más común que se le da a Blockchain es la de que “es un gran libro de acontecimientos digitales compartido colaborativamente entre muchas partes” y cumple las siguientes características:

- “Una cadena de bloques es esencialmente solo un registro, un libro mayor de acontecimientos digitales que está “distribuido” o es compartido entre muchas partes diferentes”.
- “Sólo puede ser actualizado a partir del consenso de la mayoría de participantes del sistema y, una vez introducida; la información nunca puede ser borrada”.

- “La cadena de bloques de Bitcoin contiene un registro certero y verificable de todas las transacciones que se han hecho en su historia”.

Por lo que basándonos en estas premisas sería imposible modificar un bloque ya insertado en la cadena de bloques, y para añadir un bloque “falsificado” necesitaríamos engañar a la mayoría de máquinas de la red, red que con el paso del tiempo sigue creciendo y volviéndose más compleja.

La relevancia de las monedas virtuales no reside en ellas mismas, sino en el sistema de red distribuida que hay funcionando internamente denominado Blockchain, pues no necesariamente necesita trabajar con monedas virtuales, sino que puede ser una red a disposición de cualquier tipo de datos lo que permite que se trabaje sin necesidad de ningún intermediario o grupo central que domine toda la información ofreciéndonos una plataforma libre de ataques o falsificaciones. Y toda esta red se crea mediante el consenso de miles y miles de computadores que están en la red, haciendo que la inserción de los datos en la cadena sea lo más transparente posible.

3.2. Cómo funciona

En el apartado de Crypto-currency ya hemos hablado un poco del funcionamiento que tiene Blockchain, ahora vamos a ver uno a uno los diferentes elementos que hacen posible la cadena.

3.2.1. Bloques

Cada bloque representa un conjunto de transacciones confirmadas, que se va uniendo a la cadena, está formado por:

- Un código alfanumérico (Hash) que enlaza el bloque anterior
- Las transacciones
- Otro código alfanumérico (Hash) que enlazará el siguiente bloque.

Ya hemos visto cómo se generan los códigos Hash y lo importantes que son para la cadena, pues cada bloque debe de ser validado por el resto de computadores de la red, entre ellos los mineros.

3.2.2. Mineros

El trabajo de los mineros consiste en que estos bloques sean validados, crear el hash correspondiente y unirlo al resto de la cadena de bloques.

Cuando se realizan varias transferencias, estos ordenadores las van confirmando y las van añadiendo a lo que sería el siguiente bloque de la cadena, una vez tienen un bloque completo, cogen el hash del último bloque para generar el hash del siguiente, de esta forma se asegura que el hash que se añadirá será único e intransferible (si se intentara insertar un bloque falsificado, el hash que produciría sería diferente del que debería de ir almacenado a la cadena y sería identificado como falso).

En base a las crypto-monedas, cada vez que un minero crea un hash con éxito, se le recomensa con monedas virtuales.

3.2.3. Nodos

Son los diferentes ordenadores conectados en la red de Blockchain. Cualquiera puede descargarse la cartera de Blockchain (por ejemplo, de Bitcoins) y contribuir a que la red sea más segura. Cada vez que se confirma un bloque nuevo en la red, se comunica a todos los nodos para que vuelvan a actualizar su cartera.

En el caso de que algún nodo quiera realizar una transacción, su blockchain local deberá estár actualizado y sincronizado con el resto de nodos.

3.3. Potencial

Las monedas virtuales son solo una pequeña parte de lo que puede proporcionarnos una cadena de bloque como Blockchain. Porque realmente su potencial radica en lo que se conoce como "Smart Contracts" (contratos inteligentes).

Para hacer una transacción de (por ejemplo) Bitcoins, ambas partes deben de estar de acuerdo con la transacción y, una vez realizada; se valida por cada uno de los ordenadores de la red para que la transacción sea de total confianza. Entonces, ¿por qué únicamente realizar transacciones de monedas virtuales?, ¿por qué no utilizar la red para validar diferentes tipos de contratos?

Esto es lo que se quiere realizar con los Smart Contracts, realizar transacciones de cualquier cosa sin tener que recurrir a un intermediario que ejerza un control total sobre la transacción.

Cualquier transacción online actualmente está creada para que un intermediario (es decir, un modelo centralizado) sea el que compruebe que ambas partes cumplen un contrato. El ejemplo más sencillo lo tenemos en la plataforma de pagos de Paypal, que para hacer la transacción debemos de pasar por esta plataforma para que actúe de intermediario.

Pero lo que se quiere hacer con estos contratos inteligentes, es que dada una red de Blockchain programar un contrato (no tiene que ser únicamente transferencias de dinero, puede ser de paso de información o recordatorios) en el que ambas partes están de acuerdo. Una vez introducido en la cadena de bloques será imposible de alterar (tanto por la seguridad que ofrece la red, como hemos comprobado antes como por que la red carece de intermediarios) por lo que el contrato se deberá cumplir.

Esta persistencia, la cual hace imposible que el valor añadido a la red de bloques se altere puede ser un buen punto de partida para generar una red de patentes, de forma que sea seguro registrar una patente y saber si alguien la había registrado (o no) con anterioridad.

Pongamos otro ejemplo, una empresa necesita desarrollar un producto y decide contratar a una segunda empresa para crearlo. Para la confianza de ambas partes, las empresas están de acuerdo en utilizar la red de bloques (Blockchain) para establecer el contrato, por lo que crean un contrato inteligente con los diferentes puntos que se han de cumplir (que el proyecto haya terminado, haya salido al mercado un día en concreto, …) y las dos empresas lo firman. En este momento la cadena de bloques registra el contrato, creando una cuenta neutra con el dinero, el cual se reembolsará a la segunda empresa una vez estén cumplidas todas las partes del contrato. Esta cadena de bloques da pie a que sea imposible de cambiar cualquier parte del contrato, sea neutra con ambas empresas y además aporta seguridad y anonimato por ambas partes.

Y este es un ejemplo con solo dos personas (empresas) involucradas, pero también se podría desarrollar (de hecho, se está desarrollando) una cadena de bloques para guardar información como si fuera una nube pero de manera distribuida, es decir; sin depender de intermediarios (como Mega o Dropbox) de forma que los datos guardados sean imposibles de eliminar (como pasó por ejemplo con la plataforma de Megaupload) y además mantengan la información de los usuarios bajo el anonimato.

Al ser Blockchain una red inalterable y fiable de datos, se podría utilizar para llevar la gestión contable de las empresas y gobiernos, para que se puede ver si cumplen con las normas vigentes.

Curiosamente uno de los sectores que estaban más en contra de las monedas virtuales, como es el caso de la banca; son los que están invirtiendo (y cada vez más) en conseguir cadenas de bloque propias con el que mejorar la fiabilidad de sus transacciones. El apartado de la banca lo veremos con más detalle en el siguiente punto, usos de Blockchain en el mundo real.

3.4. Usos en el mundo real

3.4.1. Bitcoin

Es la moneda virtual más conocida y es aquí donde surgió Blockchain. Hemos hablado de qué es una moneda virtual y cómo funciona, pero vamos a ver los usos que Bitcoin ha dado de Blockchain.

Bitcoin utiliza Blockchain para gestionar todas las transacciones (como hemos podido detallar anteriormente) y para ello, podemos ver la información que nos ofrece en su página web *https://blockchain.info/* .

Altura del Bloque	Antigüedad	Actas	Cantidad total enviada	Resuelto por	Tamaño (KB)
462576	14 minutes	2169	18,143.98 BTC	BTCC Pool	998.2
462575	23 minutes	1916	17,881.71 BTC	Bixin	998.01
462574	37 minutes	2396	12,505.71 BTC	F2Pool	999.99
462573	40 minutes	2473	21,333.72 BTC	BATPOOL	998.15

Para empezar, podemos ver al inicio de la página información que obtiene directamente de Blockchain, como puede ser información relevante de la cadena de bloques.

- Altura del Bloque: determina la posición que ocupa el bloque dentro de la cadena.
- Antigüedad: en intervalos de aproximadamente 10 – 13 minutos.
- Actas: número total de transacciones.
- Cantidad total enviada: Bitcoins.
- Resuelto por: Minero (o grupos de mineros) que ha (han) resuelto la transacción.
- Tamaño: tamaño que ocupa el bloque en la cadena.

Buscando más información podemos encontrar el Hash con el que se ha resuelto cada uno de los bloques.

Altura del Bloque	Hora	Resuelto por	Hash	Tamaño (KB)
462577 (cadena principal)	2017-04-19 14:57:07	BTC.TOP	00000000000000000184bd53b252c4e75692ac9f89ea94f793da1a1b09ab1776	999.1
462576 (cadena principal)	2017-04-19 14:41:37	BTCC Pool	00000000000000000001736535e13b99867266b74010d9bf024436bfb93f409b96	998.2
462575 (cadena principal)	2017-04-19 14:32:13	Bixin	00000000000000000170bb5c49ccb7895674cce5272448c18c20202639646508	998.01
462574 (cadena principal)	2017-04-19 14:18:32	F2Pool	00000000000000000091d47a37ea1eae6a9f4f39eff0da2c55428e237128f0bb	999.99
462573 (cadena principal)	2017-04-19 14:14:55	BATPOOL	00000000000000000d859978b81655328d2f497b9e674bb1495cac71bb2886e	998.15
462572 (cadena principal)	2017-04-19 13:59:49	AntPool	00000000000000000117b9b81654ca4819274d499f821b3c5b7123f8f985ec95	998.07
462571 (cadena principal)	2017-04-19 13:40:01	CANOE	00000000000000000018386d177c1374d21363a99fb313a5aaf7e6478526b0fd0	999.05
462570 (cadena principal)	2017-04-19 13:38:42	BTC.TOP	00000000000000000000191e3464c35f4774b78fcdbe7c04d050422a73aa91ca3	999.1

Incluso podemos ver dentro de un bloque todas las transferencias que se han producido.

Actas

8c6dbe1b809b0ac8fb4b5942b9c1f9b20c789e2cc078ba8b1ca2acceefada82e — **2017-04-19 14:57:07**

No Entradas (monedas recientemente obtenidos) → 1Hz96kJKF2HLPGY15JWLB5m9qGNxvt8tHJ 14.2467298 BTC

14.2467298 BTC

4ad6ddf7bbac4906b52e4554c145129012f5630addbf91be9081b74160a0a455 — **2017-04-19 14:45:30**

14RvjWBZuvKryNeipjt3BiiZG8LegCXY3c → 1JJ1VyLtexL8zCUhGiCcs1TzLc9WCKShAR 0.166718 BTC
14RvjWBZuvKryNeipjt3BiiZG8LegCXY3c 1.28030198 BTC

1.44701998 BTC

09bd32d38e846926f6032fed75999e132ef812c2d93baf0c0f389f4e77c70e24 — **2017-04-19 14:42:36**

145gDZwhHssnALUiPdEAkNFUs634riWWg2 → 1MXfJnzYUxz85ZV5K1wVKGT6qTLHJEpWU5 0.00195802 BTC
3KmDv62AgJNTr3th2Fwnm2hAEsgyFwvqQv 0.01313 BTC

0.01508802 BTC

07828221ff837024d7623945fcb5bc07f318a644e6a7a2ea43fa6bd02896cb9d — **2017-04-19 14:56:59**

1KPHtj7rKF7b3fYUVSKCLuZ1FZUzuGgGmp → 1FguTuLr3ngVaWmkuPDDs5ticEo5YADDth 0.01915095 BTC
1Jk65C4pZ9GZgXQ7YJHAGBK3FcUjTun7ZE 0.18299481 BTC

0.20214576 BTC

Pudiendo obtener información acerca de las direcciones públicas que han hecho transacciones (incluso ver su historial).

Como se puede ver, Blockchain ofrece toda la información de forma pública, para que cualquiera pueda ver el funcionamiento de las transacciones. Además de ello no solo información, Blockchain tiene una API con la que puedes crear tu propio buscador de información, e incluso crear un propio monedero.

Create Wallet API Create blockchain wallets programmatically

The create_wallet method can be used to create a new blockchain.info bitcoin wallet.

URL: http://localhost:3000/api/v2/create
Method: POST or GET

- **$password** The password for the new wallet. Must be at least 10 characters in length.
- **$api_code** An API code with create wallets permission.
- **$priv** A private key to add to the wallet (Wallet import format preferred). (Optional)
- **$label** A label to set for the first address in the wallet. Alphanumeric only. (Optional)
- **$email** An email to associate with the new wallet i.e. the email address of the user you are creating this wallet on behalf of. (Optional)

Please create an API code here including permissions to "Create Wallets".

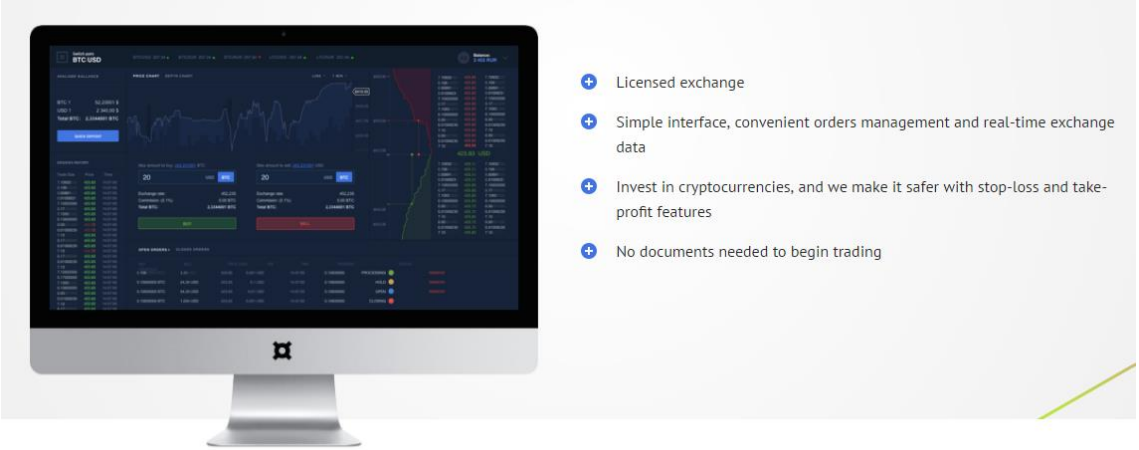
Response: 200 OK, application/json

```
{
    "guid": "4b8cd8e9-9480-44cc-b7f2-527e98ee3287",
    "address": "12AaMuRnzw6vW6s2KPRAGeX53meTf8JbZS",
    "label": "Main address"
}
```

De esta forma no solo hay un banco de bitcoins, hay diversas webs que utilizan esta API para proporcionar a sus usuarios formas de transacción y monederos. Estas son algunas de las páginas más solicitadas:

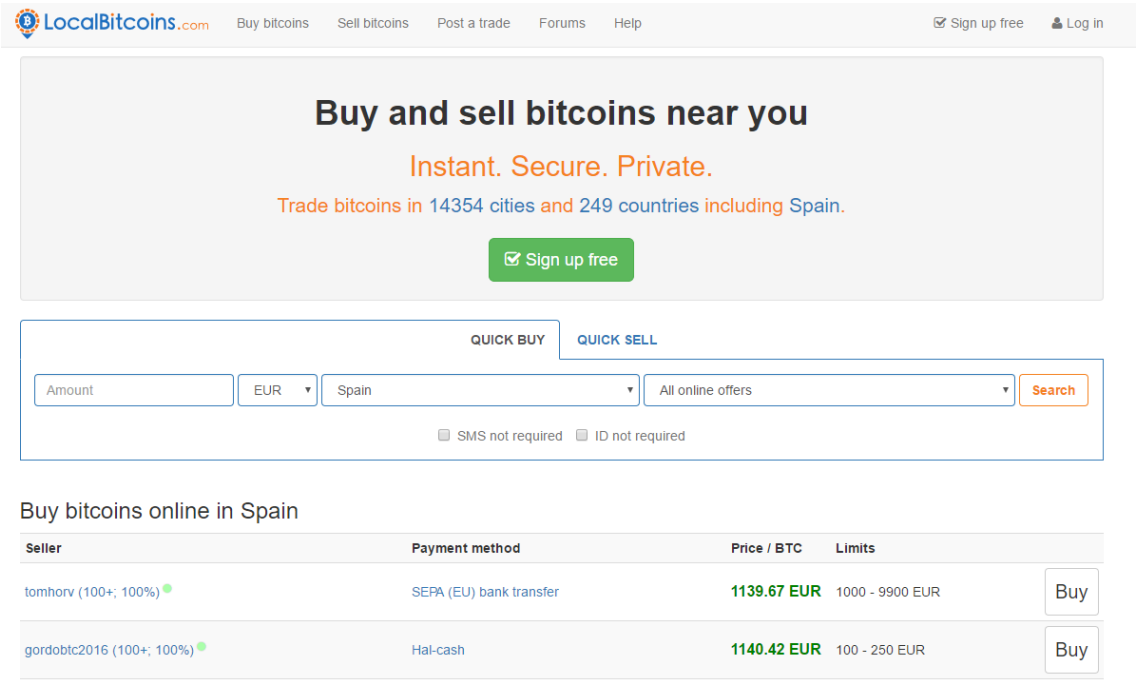
- Coinsbank

Aplicación tanto web como escritorio que te permite (además de compra / venta de Bitcoins) el cambio de Bitcoins a otro tipo de divisas.

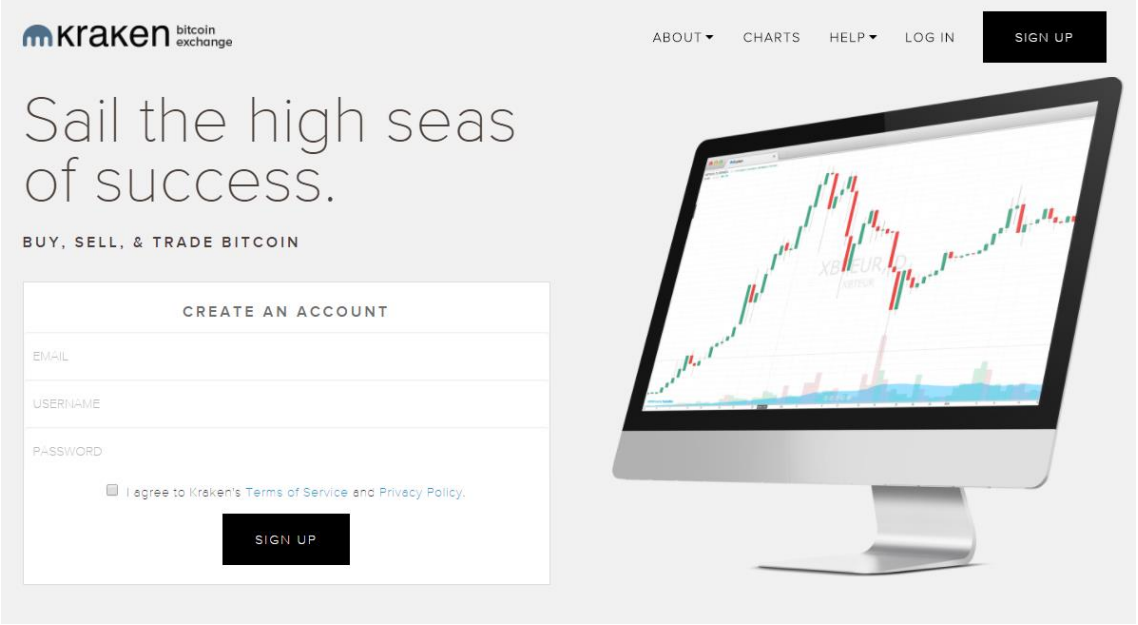


- Poloniex

- LocalBitcoins



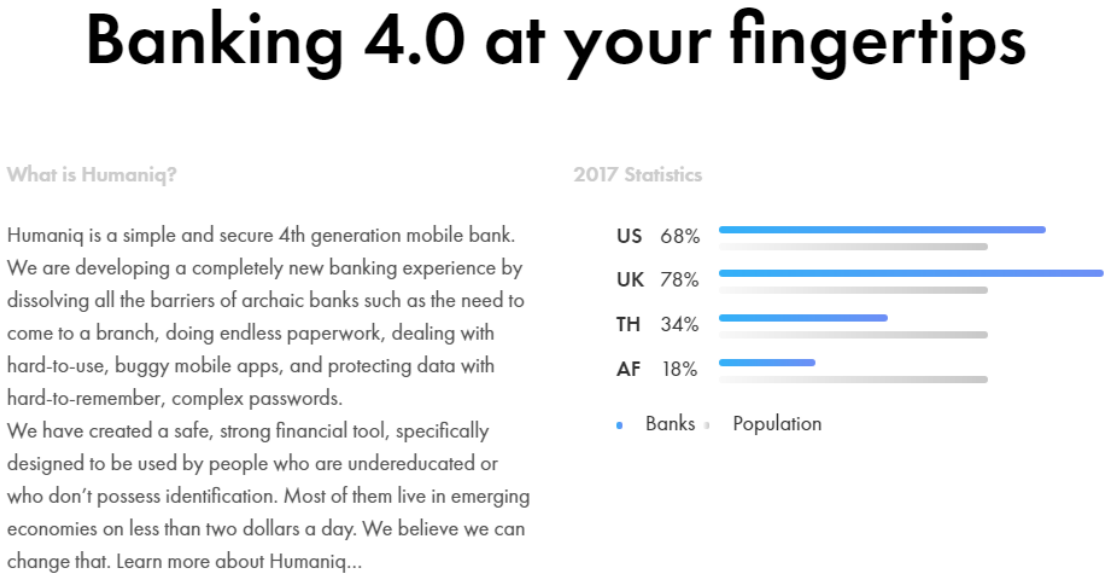
- Kraken



Y así con muchas páginas webs de diferentes nacionalidades y pensadas para diferentes públicos.

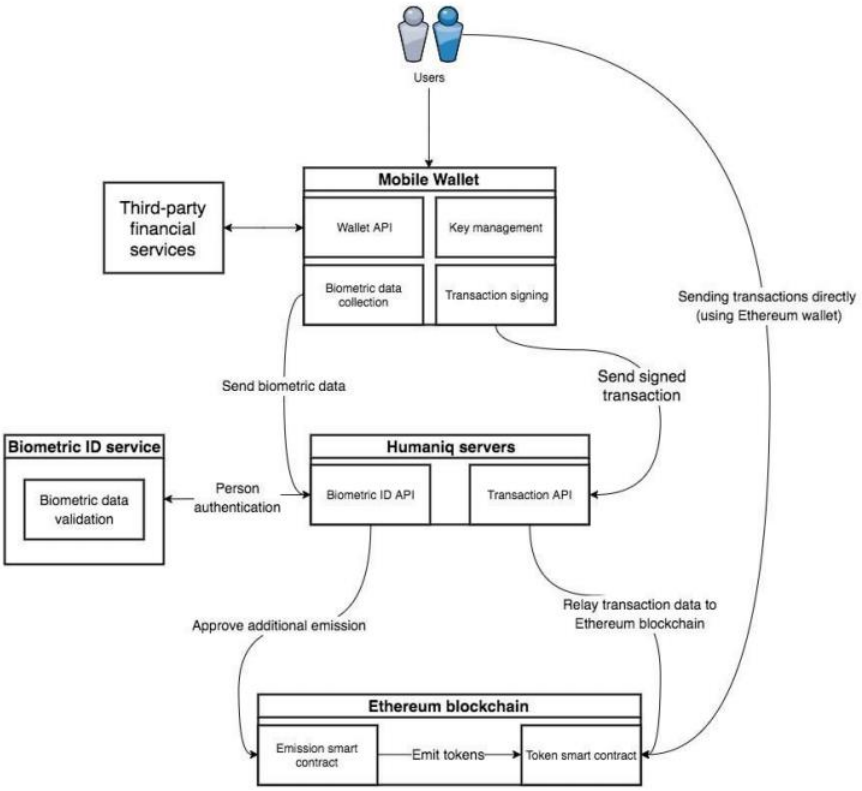
Hemos visto como Blockchain es utilizado por una moneda virtual como Bitcoin, pero vamos a ver otros métodos de uso de esta cadena de bloques.

3.4.2. Humaniq



Humaniq es un proyecto que utiliza la red de Blockchain para crear una nueva generación de servicios financieros en la que tiene como objetivo llegar a todas partes del mundo, de forma que toda esa gente que no dispone de accesos a bancos o simplemente no tiene medios para mejorar económicamente, pueda obtener liquidez para poder tener nuevas oportunidades.

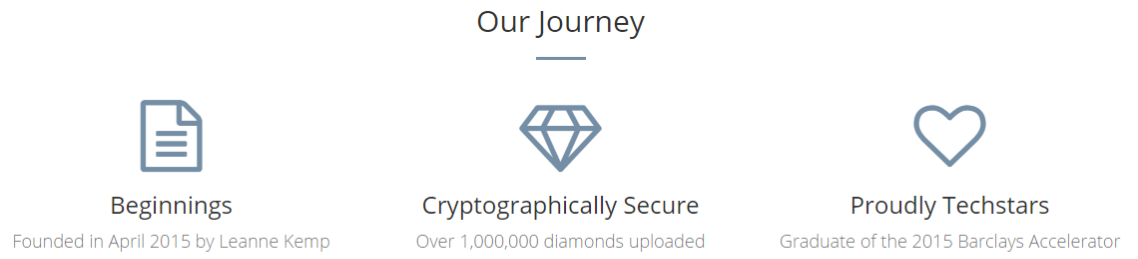
Blockchain ayuda en este proyecto a nivel de transacciones y pagos a nivel internacional puesto que no habría ni comisiones, ni terceros pudiendo realizarlas en cuestión de minutos. Además, el proyecto tiene como objetivo secundario formar a gente en estos países de pobreza económica para que entiendan como la criptoeconomía puede ayudar a su entorno.



3.4.3. Everledger

Everledger utiliza la red Blockchain para registrar productos de alto valor con el fin de reducir el fraude y robo de éstas.

Como hemos dicho, Blockchain construye una red que todo lo que se añade en forma de bloque es inalterable, por lo que el proyecto de Everledger consiste exactamente esto, registrar objetos y vincularlos a nosotros de forma que se sepa que son nuestros.



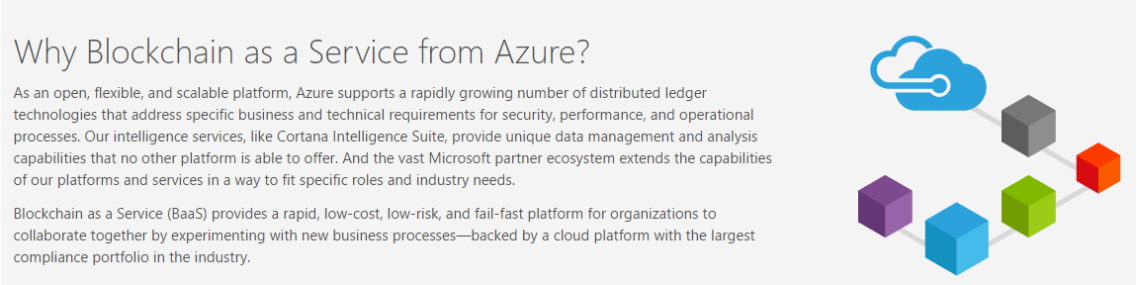
3.4.4. Bancos, aseguradoras y empresas de salud

Varias empresas de estos grupos están invirtiendo grandes cantidades de dinero en crear sus diferentes redes con Blockchain para, entre otras cosas; asegurar la confidencialidad de sus transacciones (o en el caso de pacientes, proteger la identidad y los datos personales).

- BBVA: “Uno de los primeros impactos más inmediatos sería la reducción de costes. Actualmente cada banco tiene desplegados una serie de servidores, donde la información que reside en ellos se duplica con la de otros muchos bancos e instituciones. Para actualizar la información de una base de un banco con la de otro, utilizamos procesos de mensajería”.
- Santander: “El dinero digital será la clave en el futuro de los mercados financieros y estará basado en el blockchain, una tecnología que podría revolucionar la banca en los próximos años. USC ya tiene planes para hacer pruebas en el mercado”.

3.4.5. Microsoft

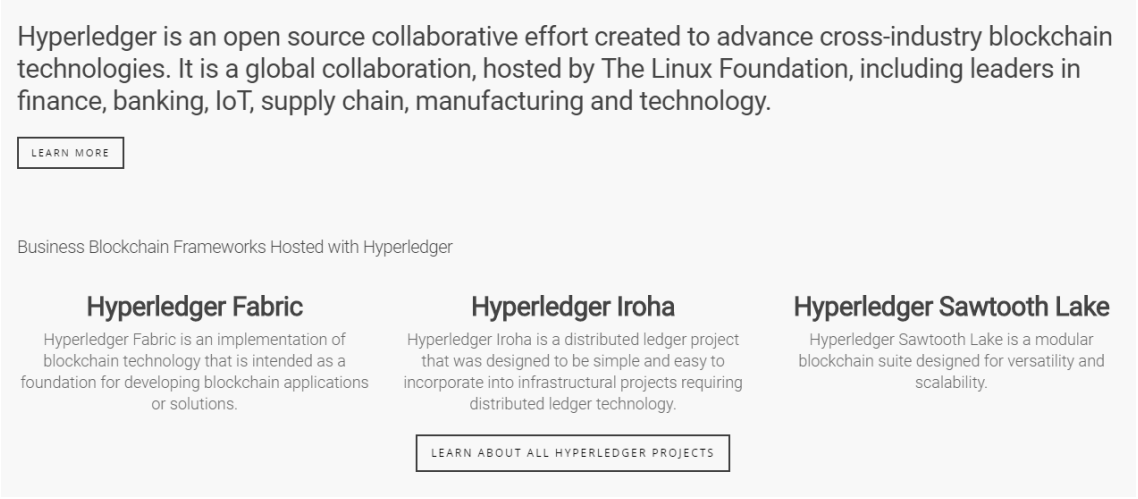
Microsoft también ha visto el potencial que tiene blockchain y desde su plataforma Azure está ayudando a generar cadenas de bloques para que otros usuarios se puedan beneficiar de ellas con lo que se conoce como BASS (Blockchain as a Service).



3.4.6. Hyperledger

Este concepto de BASS (Blockchain as a Service) consiste en ofrecer a empresas directamente la red de bloques para que éstas puedan utilizarlas para lo que quieran sin necesidad de tener que generar Blockchain desde cero.

Con esta premisa nace Hyperledger (proyecto de IBM y The Linux Foundation entre otros) en el que da una serie de herramientas open source para que las empresas puedan implementar una cadena de bloques de forma sencilla.



Y sobre este concepto de BAAS es inevitable hablar de la que hasta ahora es la red más importante de estas características, Ethereum del que hablaremos en el siguiente capítulo.

4. Ethereum

4.1. Introducción

Ethereum es la plataforma de BAAS más conocida (y utilizada) a nivel mundial. Propone una plataforma distribuida (gracias a una red Blockchain) en la que es posible escribir contratos inteligentes (aplicaciones creadas para funcionar tal y como están programadas, sin fallos, censura o fraude y, lo más importante; sin intervenciones de terceros).

Las tres premisas sobre las que se basa Ethereum son:

- Las aplicaciones Ethereum siempre ejecutan el código que dicen que ejecutarán.
- Las aplicaciones Ethereum siempre están disponibles.
- Las aplicaciones Ethereum son resistentes a ataques.

Bajo el concepto de estas aplicaciones, y la seguridad que tiene la red distribuida de Blockchain, estamos creando contratos seguros e inalterables, basándonos en las "leyes" que nosotros añadamos a dicho contrato.

Para crear estos contratos, Ethereum se basa en dos partes, la primera de ellas es mediante su propia moneda virtual llamada Ether y la segunda mediante su propio lenguaje de programación llamado EtherScript.

- Ether

Moneda virtual de la plataforma Ethereum que funciona igual que la moneda Bitcoin pero con propósitos diferentes.

De la misma manera que para Bitcoin los mineros son necesarios para crear los bloques con las diferentes transacciones, en Ethereum los mineros ayudan para crear los bloques de los contratos que se van añadiendo. De esta forma, cada minero que completa un bloque con éxito recibe una recompensa en forma de moneda virtual llamada Ether (al igual que pasa con la plataforma Bitcoin).

La diferencia que tiene con Bitcoin, es que mientras una está pensada para ser una moneda con valor en el mundo real, Ether está creada con el propósito de utilizarse en la creación de contratos (aunque esto no quita que, si hay inversores invirtiendo en esta moneda; pueda darse que se consolide como un valor a tener en cuenta).

Por lo tanto, cuando se crea un contrato, antes de poder subirlo a la red hay que hacer un pago en Ethers, de esta forma Ethereum se evita de que se puedan añadir contratos maliciosos (dado que por cada uno que se añada, habría que pagar una cantidad de Ethers) y que el código de estos contratos sea eficiente (Ethereum detecta si es eficiente o no el código programado y aumenta la cantidad de Ethers a pagar en función a esto).

- EtherScript

Es un lenguaje de programación propio de la plataforma Ethereum con el cual se programan los contratos mediante unas instrucciones muy precisas. Gran parte de los contratos incluyen un

pago y una fecha determinada, con lo que este lenguaje está diseñado para generar de forma fácil (y entendible por ambas partes) este tipo de contratos.

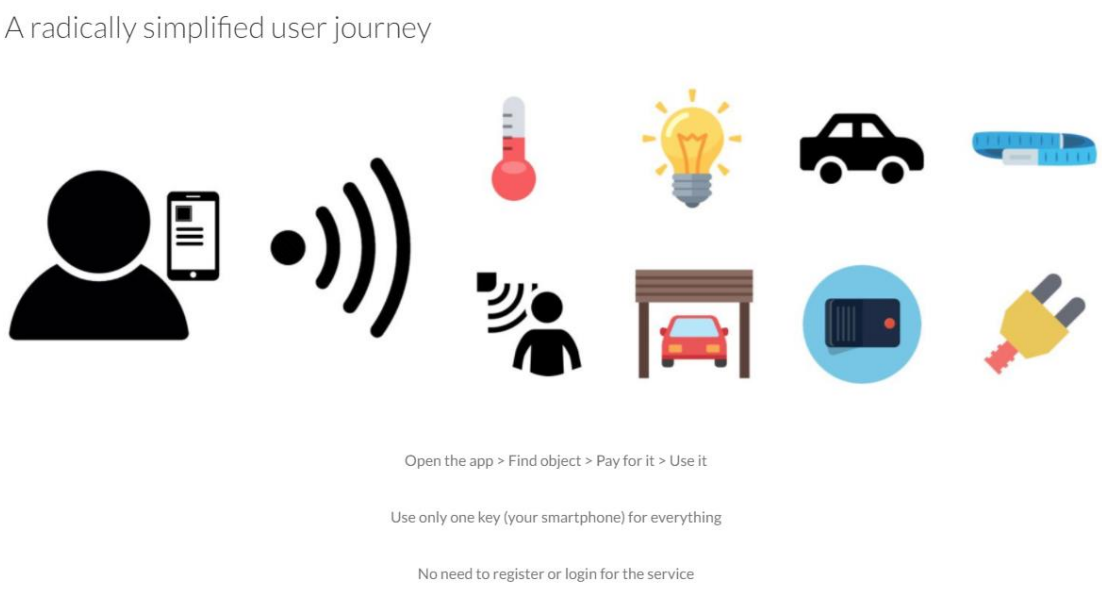
Una vez creado el contrato y todas las partes están de acuerdo, cuando entra en la cadena de bloques ya está sujeto bajo las condiciones de la red de forma que este contrato permanece inalterable.

4.2. Usos en el mundo real

4.2.1. Slock.it

Consiste en un ambicioso proyecto en el que quiere combinar la red de Blockchain con IoT (Internet of Things) de forma que cualquier objeto registrado en la red se puede alquilar.

La red que están construyendo se llama USN (Universal Sharing Network) y permite al usuario encontrar en una zona en concreto alquilar vienes (por ejemplo, un vehículo o una habitación).



Para ello, están desarrollando (bajo Ethereum) una red denominada “Ethereum Computer” en la que animan a la gente a seguir desarrollando (el proyecto es Open Source).

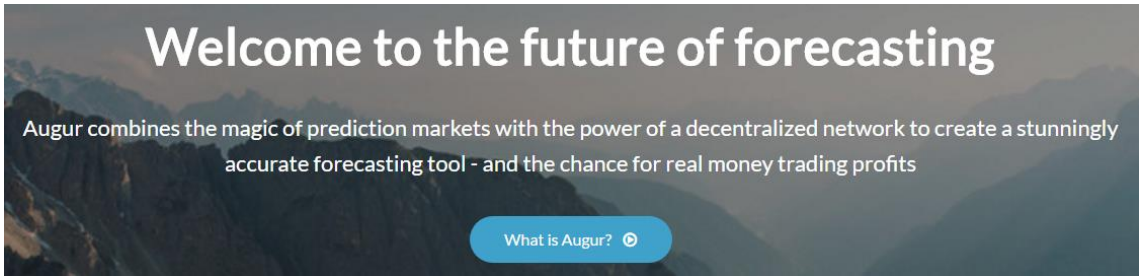
4.2.2. Augur

Augur es un sistema de predicción de mercados en el que, en lugar de ser una persona (o grupos de personas) quienes hagan estas predicciones; es la red de bloques (Blockchain) quien hace la tarea.

Como cualquier otra plataforma de este estilo, lo que pretende Augur es ir un paso por delante de lo que puede pasar por el mercado financiero para que la gente pueda invertir en los sitios correctos justo antes de que pase.

El hecho de que hayan querido hacer un sistema de predicciones descentralizado es por la idea de que, por lo general; las predicciones suelen venir de un mismo sitio, una misma organización o una misma persona (información centralizada) por lo que la información que nos provee puede estar manipulada a su favor.

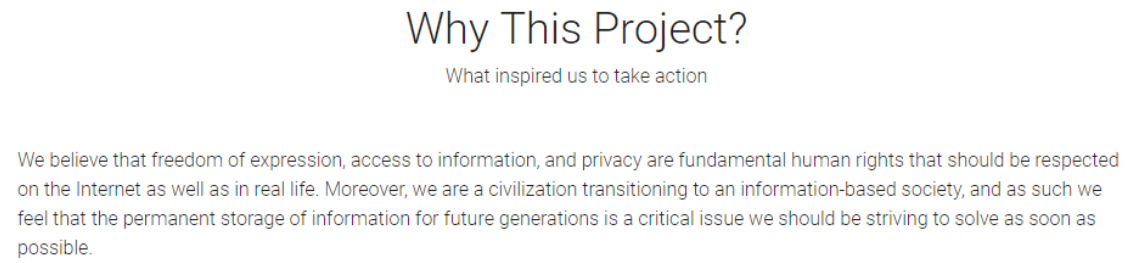
Por ello, gracias a esta red descentralizada de predicciones, podemos saber que la información que nos viene nunca estará manipulada por terceros.



4.2.3. Akasha

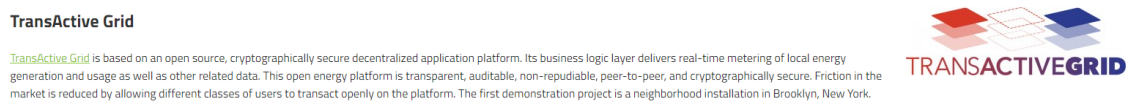
La idea de Akasha parte de crear una red social distribuida de forma que sea capaz de crear agrupaciones con gente que tengan las mismas cosas en común basado en un sistema de clasificación inteligente.

Mediante la red de Ethereum lo que se quiere realizar son publicaciones basadas en contratos de modo que esté bien definido lo que se puede censurar y lo que no.



4.2.4. LO3

LO3 es una empresa energética que entre sus proyectos está TransActive Grid, el cual se beneficia de la plataforma Ethereum para definir contratos de compra y venta de energía entre consumidores.



5. Y esto es sólo el principio

La tecnología de las monedas virtuales (y de Blockchain en general) tiene apenas unos 10 años, y poco a poco las empresas están viendo todo su potencial en base a la red de datos que es capaz de generar.

Cada vez más se ven más noticias de cómo las empresas invierten dinero en crear sus propias redes de Blockchain con el fin de tener su información segura, fiable y sin riesgos de terceros.

Y por supuesto, es sólo el principio dado que las ideas que se intentan llevar a cabo están en su fase más temprana pero ya hemos podido ver el enorme potencial que puede suponer una red de estas características a nivel mundial.

6. Simulación de una moneda virtual utilizando Blockchain

6.1. Introducción

Este proyecto consiste en una emulación de cómo funciona una moneda virtual con una red de bloques, desde la creación de un Blockchain propio hasta el minado y las transferencias de las monedas. Dado que el funcionamiento de una cadena de bloques es mucho más complejo de lo que pudiera utilizar el proyecto, se ha optado por generar una simulación de éste de forma que podamos comprender como funciona interiormente sin necesidad de tener una complejidad tan alta.

A la hora de pensar cómo afrontar la creación de una moneda virtual observé que en todas las monedas había varios puntos en común que debía de tener en cuenta.

- Libro de cuentas: utilizando una red de bloques parecida a Blockchain como lugar donde almacenar la información de todas las transferencias.
- Minero: para comprobar las transacciones y así poder insertarlas en la red de bloques.
- Lugar de transferencia: un sitio dónde los usuarios puedan crear sus propias carteras y hacer transferencias entre ellos.

Por lo que la estructura pensada para este proyecto constaría de tres partes totalmente independientes, por un lado el lugar donde realizar las transferencias (mediante página web, como si de un banco de Bitcoins se tratara), por otro un libro de cuentas donde almacenar la información de las transferencias y, finalmente; un minero simulando la búsqueda de un bloque y confirmando la transferencia en el libro de cuentas.

6.2. Realización del proyecto

6.2.1. Metodología

Tal y como había planteado el proyecto era fácilmente divisible en tareas (bloques) en los que fuera iterando para tener cada cierto tiempo un “entregable” con el que poder mostrar mis avances. Además de ello, el proyecto en sí estaba dividido en tres partes (Blockchain, Web de la moneda virtual y Minero) por lo que podía trabajar en ellos de manera independiente, por lo que, pese a ser solo una persona (es decir, tenía el rol de Scrum Master, Product Owner, desarrollador, …) traté de imponer en mi forma de trabajo una parte de la metodología ágil que iba dedicada a los entregables.

6.2.2. Plan e Iteraciones

Por motivos laborales tuve que concentrar la mayor parte del trabajo los sábados y domingos, por lo que las iteraciones del proyecto tienen diferentes tiempos (algunas iteraciones son de 2 semanas, otras de 1 mes, …) y algunas incluso se hicieron en paralelo.

Las primeras iteraciones fueron más de investigación y las finales fueron más a nivel de testing.

El plan asociado a los proyectos ha sido el siguiente:

- Iteración 1
    - Búsqueda de información acerca de Bitcoin.
    - Cómo surgió, causa de éxito.

- Iteración 2
    - Creación de una cartera de Bitcoin.
    - Descarga de la red Blockchain y comprender su uso.

- Iteración 3
    - Investigación sobre diferentes monedas virtuales.
    - Comprender los puntos en común.

- Iteración 4
    - Investigación sobre el minado de Bitcoins.
    - Comprender cómo se generan los Bitcoins.

- Iteración 5
    - Análisis del proyecto, puesta a punto.
    - Definir los proyectos independientes (Minería, Red de Bloques y Banco).

- Iteración 6
    - Investigación sobre Blockchain.
    - Comenzar el desarrollo del proyecto Blockchain.
        - Definir el lenguaje.
        - Definir la arquitectura.
            - Investigar sobre arquitecturas N-Capas.
        - Definir los modelos de base de datos.

- Iteración 7
    - Continuación del proyecto Blockchain.
        - Continuar definiendo la arquitectura.
        - Desarrollar los casos de uso y el diagrama de base de datos.
        - Definir las diferentes entidades.

- Iteración 8
    - Continuación del proyecto Blockchain.
        - Definir los diferentes servicios web.
            - Definición de un servicio web para el Banco.
            - Definición de un servicio web para el Minero.
        - Crear la base de datos.

- Iteración 9
    - Continuación del proyecto Blockchain.
        - Crear la capa de persistencia.
            - Crear los repositorios.
        - Crear la capa de dominio.
            - Crear las entidades de Entity Framework.
            - Definir las interfaces de la capa de persistencia.

- Iteración 10
    - Investigación sobre el funcionamiento de las webs de bancos de Bitcoin.
    - Comenzar el desarrollo del proyecto Banco.
        - Definir el lenguaje.
        - Definir la arquitectura (Arquitectura N-Capas).
        - Investigar sobre la capa de Presentación.
            - Definir un proyecto de consola para las pruebas.
            - Definir un proyecto de web para la parte de User Interface.

- Iteración 11
    - Continuación del proyecto de Blockchain.
        - Crear inyección de dependencias con Unity.
        - Crear la capa de aplicación.
            - Crear las interfaces.
            - Crear las clases.
        - Definir los métodos de la capa de dominio.
            - Mapear la transacción que nos venga de la web del Banco a un DTO de Blockchain.
            - Crear un bloque dentro de Blockchain.

- Iteración 12
    - Continuación del proyecto de Blockchain.
        - Crear los métodos de la capa de dominio.
        - Crear un proyecto de consola para mapear Blockchain.

- Iteración 13
    - Continuación del proyecto de Blockchain.
        - Bugfixing.
        - **Primer entregable de Blockchain.**

- Iteración 14
    - Continuación del proyecto Banco.
        - Crear la Base de Datos.
        - Crear la capa de Dominio.
            - Añadir las entidades (Entity Framework).
            - Añadir las interfaces de los repositorios.
        - Crear la capa de persistencia.
            - Crear los repositorios.

- Iteración 15
    - Continuación del proyecto Banco.
        - Crear el inyector de dependencias (Unity).
        - Crear la capa de aplicación.
        - Crear el proyecto de consola.

- Probar que la capa de aplicación y la capa de persistencia funcionan correctamente.

- Iteración 16
    - Continuación del proyecto Banco.
        - Definir y crear las llamadas al servicio web de Blockchain.
        - Probar las llamadas al servicio web de Blockchain.
        - Comprobar que se crean correctamente los Bloques y su respectivo Hash.

- Iteración 17
    - Continuación del proyecto Blockchain.
        - Refactorizar para combinarlo con el proyecto Banco.
        - Crear una cola para simular las peticiones de los bloques.
        - **Segundo entregable de Blockchain.**

    - Continuación del proyecto Banco.
        - Investigar acerca de MVC (Model View Controller).
        - Investigar como generar las diferentes Vistas.
        - Definir arquitectura de la capa de presentación.

- Iteración 18
    - Continuación del proyecto Banco.
        - Creación del patrón MVC.
            - Crear los Modelos.
            - Crear las Vistas / Partial Views.
            - Crear los Helpers.
            - Crear los controladores.

- Iteración 19
    - Continuación del proyecto Banco.
        - Refactorizar las vistas.
        - Continuar la creación de los controladores.

- Iteración 20
    - Continuación del proyecto Banco.
        - Probar el proyecto.
            - Comprobar que la interfaz de usuario se conecta correctamente con el resto del proyecto.
            - Comprobar la conexión del servicio web con Blockchain.
            - Comprobar que las vistas sean correctas.
        - **Primer entregable del proyecto Banco.**

- Iteración 21
    - Definición del proyecto Minero.

    - Crear el proyecto Minero.
        - Programa sencillo que escucha el servicio web de Blockchain.
        - Comprueba que hay transacciones nuevas, ve que son correctas y genera una petición para crear un bloque en Blockchain.
        - **Primer entregable del proyecto Minero.**

- Iteración 22
    - Pruebas del entorno con los tres proyectos.
    - Refactorización y bugfixing.
        - **Tercer entregable del proyecto Blockchain.**
        - **Segundo entregable del proyecto Banco.**
        - **Segundo entregable del proyecto Minero.**

- Iteración 23
    - Pruebas con mayor cantidad de datos.

6.2.3. Selección del lenguaje C#

Al inicio de la definición del proyecto, vi que iba a necesitar un lenguaje robusto diseñado para crear grandes soluciones, como vi que al menos de las 3 soluciones, dos iban a ser grandes (el proyecto de Blockchain y el proyecto del Banco) me decanté por C#.

C# (y .Net en general) me permitía un amplio abanico de soluciones para proyectos grandes, y una amplia selección de arquitecturas y patrones tanto para backend, front end y servicios web por lo que la definición de la arquitectura iba a ser enfocada en este lenguaje.

Durante el proyecto he visto con qué facilidad me permitía el lenguaje utilizar patrones SOLID, añadir la base de datos con Entity Framework y utilizar el modelo de MVC para la vista de la web.

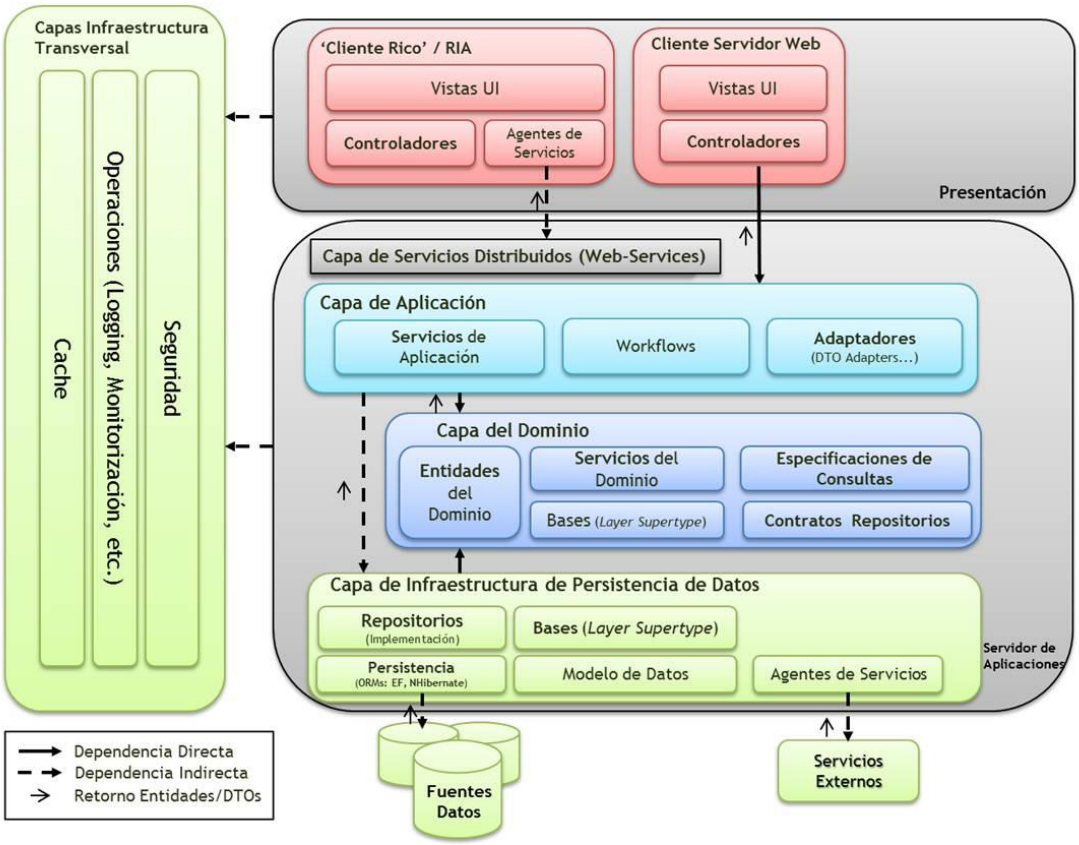
6.2.4. Arquitectura N-Capas

A la hora de definir la arquitectura para los proyectos de Blockchain y el Banco, observé que iba a necesitar dar una solución para las siguientes estrategias:

- Proporcionar una base de datos y sus entidades orientadas a objetos.
- Obtener la información necesaria de las transacciones, actualizarla y volver a añadirla.
- Inyección de dependencias para las diferentes interfaces del proyecto.
- Acceso a servicios web para que los diferentes proyectos pudieran comunicarse.
- Vista para las consultas del cliente mediante la web.
- Proporcionar un fácil manejo de la información para poder utilizarla en los diferentes ámbitos del proyecto (DTOs)
- Cumplir con los patrones SOLID para que la solución sea mantenible en cuanto a cambios.

Debido a que las soluciones iban a ser grandes, necesitaba una arquitectura que me diera solución a estas preguntas por lo que la solución adecuada fue utilizar el modelo N-Capas.

Arquitectura N-Capas con Orientación al Dominio



El modelo N-Capas es utilizado (en su mayoría) para ofrecer una arquitectura a proyectos grandes, en los que estos necesiten utilizar diferentes enfoques (desde proporcionar una vista al usuario hasta utilizar servicios web, moldear la información y guardarla en base de datos).

La idea de este modelo es crear una arquitectura en la que se vayan añadiendo capas con el fin de abstraer todas las soluciones del proyecto. La explicación de cada capa es la siguiente:

- Capa de Presentación

Es la parte más visual del proyecto y es donde se definen las diferentes soluciones para las vistas de la interfaz de usuario, esta capa recoge la información de la capa de aplicación. Gracias a esta capa podemos tener diferentes vistas (por ejemplo, una vista web y una vista de consola) utilizando los mismos datos sin necesidad de hacer grandes cambios.

- Capa de Aplicación

La capa de aplicación es la que transforma la información (las diferentes entidades y modelos) tanto de la capa de persistencia como de la capa de dominio para finalmente enviarla a la capa de presentación. De esta forma, solo enviamos a la capa de presentación únicamente la información que necesite sin necesidad de duplicar código. Para ello, en la capa de aplicación se definen los diferentes DTOs para transformar esta información.

- Capa de Domino

Por una parte, se encarga de crear los contratos (interfaces) de los diferentes repositorios de la capa de persistencia y por otra, se encarga del manejo de las operaciones de la aplicación (dominio de la aplicación) para enviarlo a la capa de aplicación. Es la capa que se encarga de las operaciones más complejas a nivel de Backend.

- Capa de persistencia

Es la capa a más bajo nivel y la que accede a base de datos mediante repositorios, por lo general cada uno de los repositorios accede a una entidad diferente mediante las operaciones CRUD.

- Capa transversal

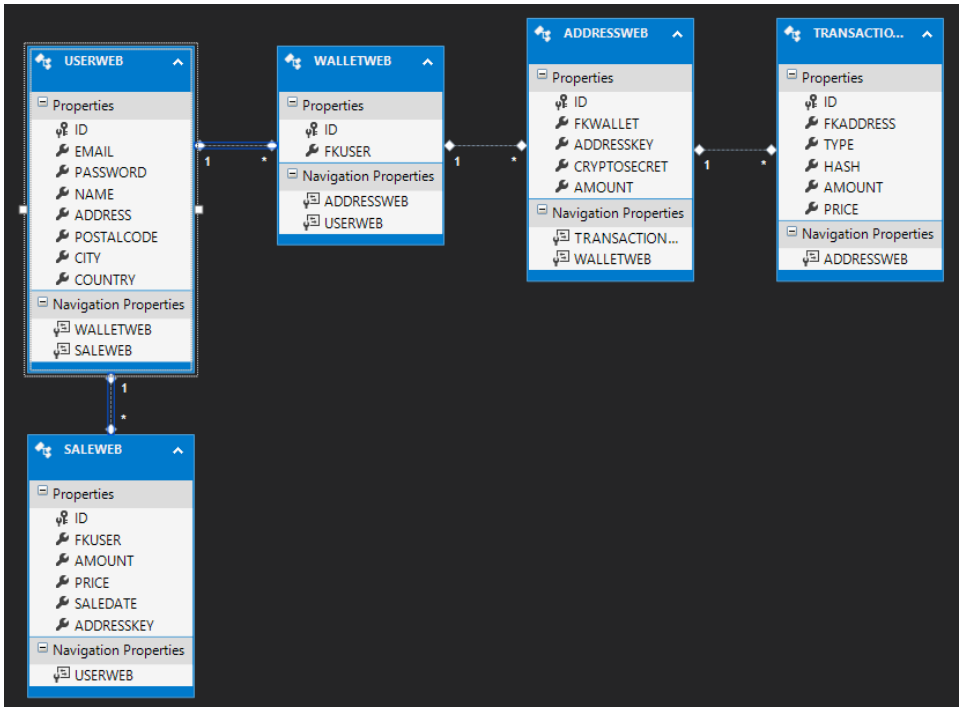
Capa auxiliar en la que se añade la lógica que es utilizada por toda la aplicación, que va desde el log y la monitorización hasta crear la inyección de dependencias.

- Capa de servicios distribuidos

Se encarga de la gestión de los diferentes servicios web del proyecto.

6.2.5. Entity Framework

Dado que los proyectos requerían una fuerte estructura de acceso a datos, se ha optado por utilizar lo que se conoce como Entity Framework. Gracias a ello es posible generar las tablas de la base de datos y de forma automática las transforma a entidades dentro del proyecto (también se puede hacer a la inversa, con el método conocido como code first). De esta forma únicamente nos debemos de preocupar en crear desde un principio una estructura de base de datos sólida, la cual se encargará de transformar a entidades Entity Framework. Aquí un ejemplo de cómo mapea una base de datos.



6.2.6. Inyección de Dependencias con Unity

La inyección de dependencias es uno de los patrones más importantes a la hora de hacer cualquier tipo de desarrollo. La función de la inyección de dependencias es la de separar el código por responsabilidades, es decir que la parte de la creación del objeto y la parte de la inicialización de este deben de ir separados.

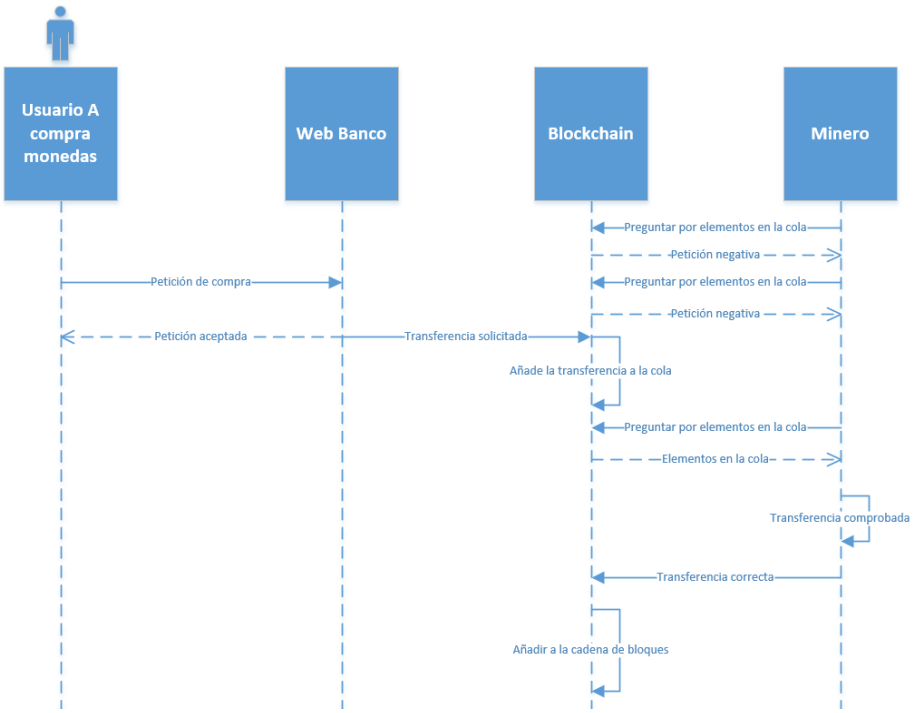
Para ello, utilizamos un contenedor de dependencias (en este caso Unity) que nos ayuda a generar los diferentes objetos dependiendo de la interfaz que inyectemos, de esta forma únicamente nos tenemos que preocupar de inyectar la interfaz del objeto en los lugares que se van a utilizar (ya sea mediante el constructor o mediante el método).

La inyección de dependencias también nos ayuda a mejorar la sostenibilidad y la mantenibilidad de nuestra solución ya que, si necesitamos hacer modificaciones en alguna clase; basta que las hagamos en la interfaz para que el contenedor de dependencias se encargue de gestionar los diferentes comportamientos.

También nos es muy útil para hacer de las clases elementos que se puedan testear (ya sea por inyección de un stub o de un mock) y es que una de las reglas más importantes del patrón SOLID es la que cuenta que si una clase no se puede testear, directamente sabemos que no cumple este patrón.

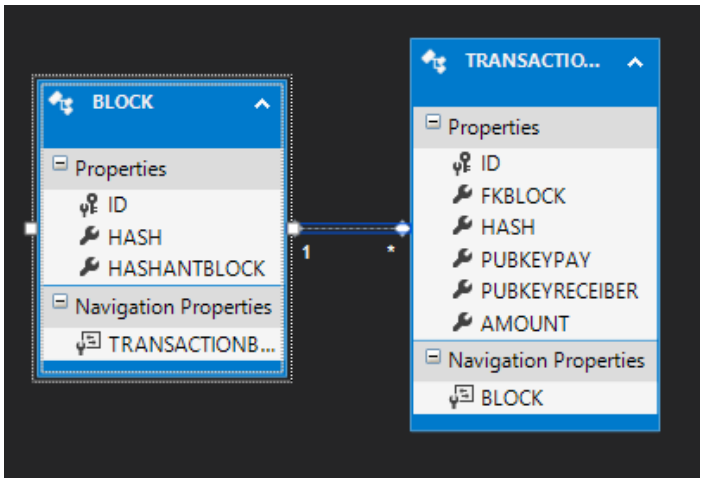
6.3. Esquema General

La estructura del proyecto consta de las tres partes ya mencionadas que son las de Minería, Blockchain y Banco (Web). La interacción entre ellas es mediante servicios web en la que, cuando un usuario realiza una transacción por la web, se comunica al proyecto de Blockchain para que a su vez la añada en cola para que el proyecto de Minería pueda comprobar que es correcta. Finalmente, se añade como bloque al proyecto de Blockchain.



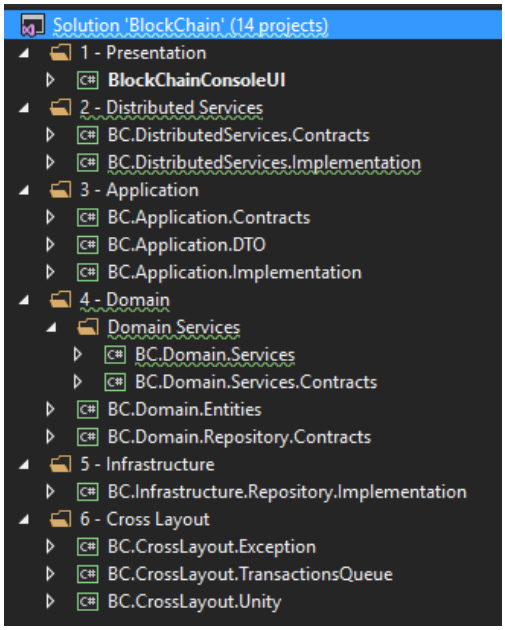
6.4. Proyecto: Blockchain

El proyecto de Blockchain consiste en una simulación de la cadena de bloques, en él tenemos nuestro "libro de cuentas" de todas las transacciones. Básicamente consiste en los diferentes bloques que contienen un hash, el hash del bloque anterior y una referencia a la transferencia realizada (con las claves públicas de la transferencia y la cantidad total transferida).



Un bloque puede contener varias transacciones, pero una transacción solo puede estar en un solo bloque. De esta manera nos aseguramos de que cada transferencia es única.

El proyecto, como hemos comentado antes sigue una arquitectura de N-Capas con la siguiente estructura.



La capa de Infraestructura contiene los diferentes repositorios (en este caso dos, el repositorio del bloque y el de la transacción del bloque) con las diferentes operaciones CRUD, como podemos ver utilizamos interfaces para la posterior inyección de dependencias.

```
/// <summary>
/// Clase RepositoryBlock encargada de las operaciones CRUD con la entidad <see cref="BLOCK"/>
/// <para>Implementa la interfaz <see cref="IRepositoryBlock"/></para>
/// </summary>
2 references
public class RepositoryBlock : IRepositoryBlock
```

La capa de dominio contiene los contratos de los repositorios (las interfaces), las entidades y los servicios de dominio que tiene nuestra aplicación blockchain. Tendremos únicamente dos que serán convertir un objeto a una entidad que se pueda guardar en base de datos y otro servicio que será el encargado de crear un bloque.

La capa de aplicación la utilizamos para la gestión entre la capa de dominio y la capa de persistencia.

La capa de servicios distribuidos es la que tiene los servicios web para obtener una transacción de la web, enconlar dicha transacción y finalmente añadirla en el bloque. Estos servicios no los utiliza Blockchain por sí solo, sino que son los respectivos proyectos (tanto el proyecto web como el proyecto minero) los que utilizan estos servicios.

La capa de "Cross" es la capa transversal en la que tenemos las excepciones creadas por nosotros y el inyector de dependencias.

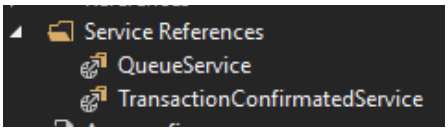
Finalmente, aunque el proyecto no lo requiera; tiene una capa de presentación que ha sido utilizada para crear una solución de consola para poder probar todo el sistema de Blockchain.

6.5. Proyecto: Minero

Este proyecto es el encargado de obtener la cola de peticiones de transferencias de Blockchain, confirmarla y añadirla a la cadena de bloques. Es una aplicación muy sencilla que únicamente consiste en una aplicación de consola con dos de los tres servicios web de Blockchain (QueueService y TransactionConfirmatedService) que está siempre escuchando peticiones.

```
namespace MainConsoleUI
{
    0 references
    class MainConsoleUI
    {
        0 references
        static void Main(string[] args)
        {
            while (true)
            {
                Thread.Sleep(TimeSpan.FromSeconds(3));

                QueueServiceClient serviceClient = new QueueServiceClient();

                if (serviceClient.GetTransactionqueue().Count > 0)
                {
                    TransactionConfirmatedServiceClient confirmatedService = new TransactionConfirmatedServiceClient();

                    Console.WriteLine("Transaccion confirmada");

                    QueueService.TransactionDTO transactionQueue = serviceClient.GetTransactionqueue().Dequeue();

                    TransactionConfirmatedService.TransactionDTO transactionConfirmatedService = new TransactionConfirmatedService.TransactionDTO
                    {
                        HASH = transactionQueue.HASH,
                        PUBKEYPAY = transactionQueue.PUBKEYPAY,
                        PUBKEYRECEIBER = transactionQueue.PUBKEYRECEIBER,
                        AMOUNT = transactionQueue.AMOUNT
                    };

                    confirmatedService.MinerTransactionConfirmated(transactionConfirmatedService);
                }
                else
                {
                    Console.WriteLine("No hay elementos en la cola");
                }
            }
        }
    }
}
```

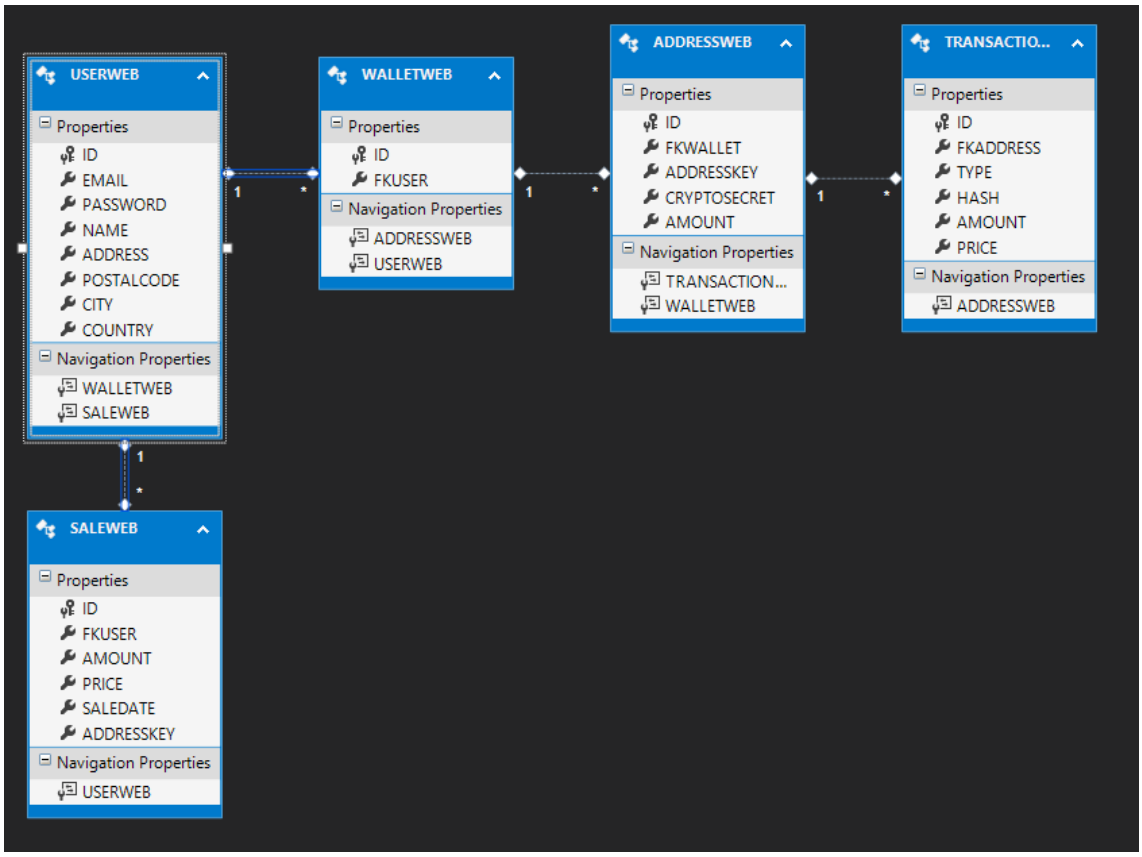
Los diferentes servicios web que utiliza de Blockchain.



6.6. Proyecto: Web Banco

Al igual que cualquier otra moneda virutal, la parte de Blockchain es accesible desde cualquier parte (normalmente a través de una API que tiene integrada la propia red de bloques) a través de nuestra moneda virtual también podemos crear una web que haba de banco para crear las diferentes transacciones.

La base de datos contiene las siguientes entidades.

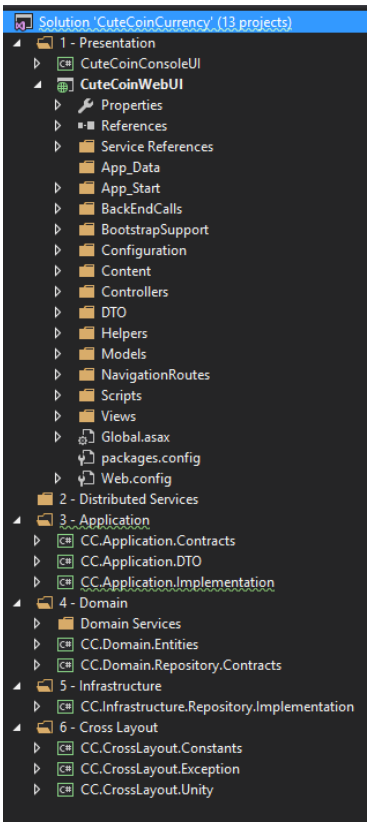


El núcleo de la web se basa en los usuarios, por lo que la base de datos gira en torno a estos. Cada usuario puede tener varios “Saleweb” que con las monedas que tiene en venta (como podemos ver, están la cantidad de monedas, el precio a la venta, la dirección pública, ….)

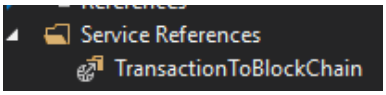
Luego para controlar la cartera del usuario, cada cartera puede contener varias cuentas diferentes (al igual que pasa con monedas virtuales de verdad, que cada usuario puede tener diferentes cuentas) en la que tenemos la cantidad de monedas, la clave pública y el “cryptosecret” que sería nuestra clave privada.

Después cada cuenta tiene las distintas transacciones (con tu tipo, el hash, la cantidad y el precio).

Al igual que en Blockchain, se ha decidido utilizar una arquitectura N-Capas que funciona de la misma manera que la que tenemos en el anterior proyecto. Lo que se diferencia con el otro es en que este tenemos la parte web de interfaz de usuario hecho con MVC.

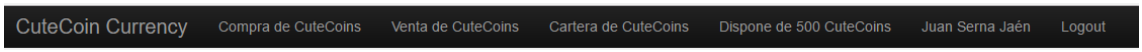


En este caso tenemos un servicio web del proyecto de Blockchain para que cuando se realice una transacción se le notifique a la cadena de bloques.



En cuanto a la web, los servicios ofrecidos son los siguientes:

- Crear Usuarios.
- Un usuario puede crear hasta 5 direcciónes diferentes.
- Un usuario puede enviar monedas a otros usuarios mediante su dirección pública.
- Un usuario puede vender monedas a un precio en concreto.
- Un usuario puede comprar monedas.



Todo ello bajo el proceso de utilizar la red de bloques por debajo, para poder recrear una simulación del comportamiento de una moneda virtual con Blockchain.

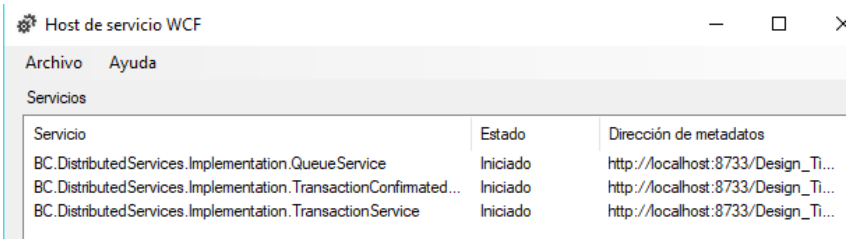
6.7. Ejemplo de uso

Vamos a generar varias operaciones a través del banco y comprobaremos como funciona realmente nuestro Blockchain simulado.

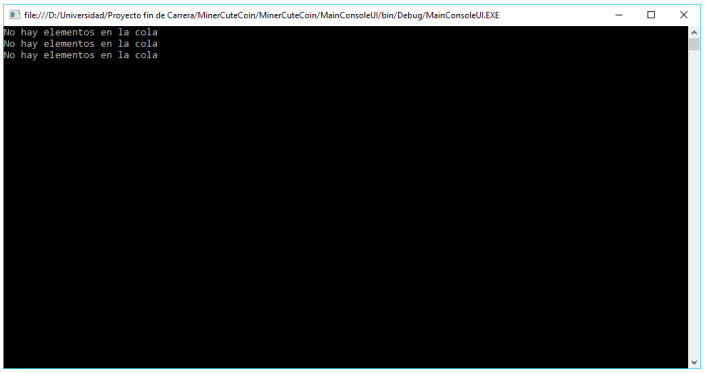
Para ello, primero de todo vamos a arrancar los tres proyectos (proyecto minero, proyecto blockchain y proyecto banco) para poder comenzar.

Al operar de forma individual cada uno, podríamos crear (por ejemplo) otro banco diferente que se conectara a la misma red de bloques o incluyo hacer diferentes aplicaciones (escritorio y móvil) que hagan uso de nuestro Blockchain.

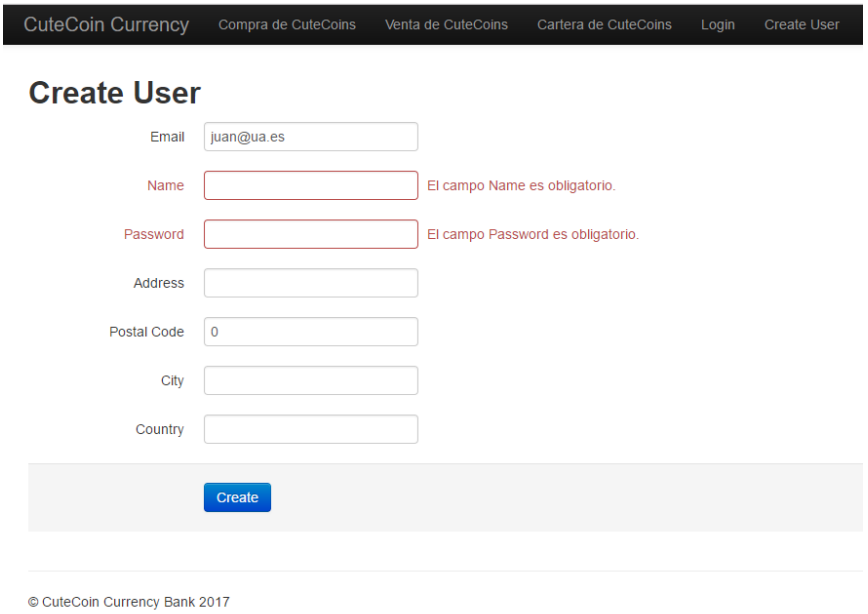
Cuando arrancamos Blockchain, se activan los tres servicios que ofrece.



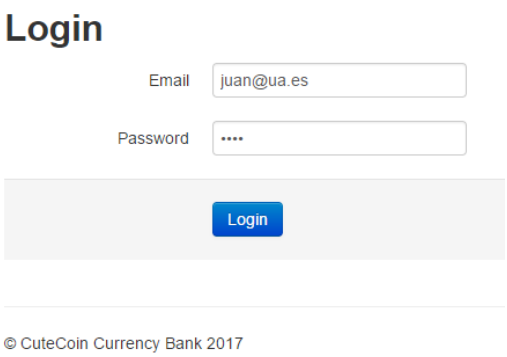
Arrancamos el minero y vemos como va solicitando peticiones a Blockchain para ver si tiene alguna transacción en cola.



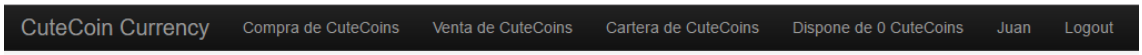
Arrancamos la web del banco y creamos un usuario, como podemos ver algunos campos son obligatorios.



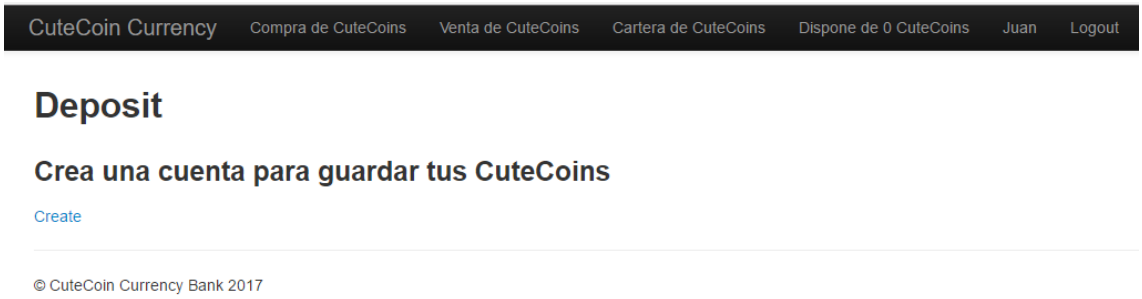
Creamos el usuario y nos logueamos.



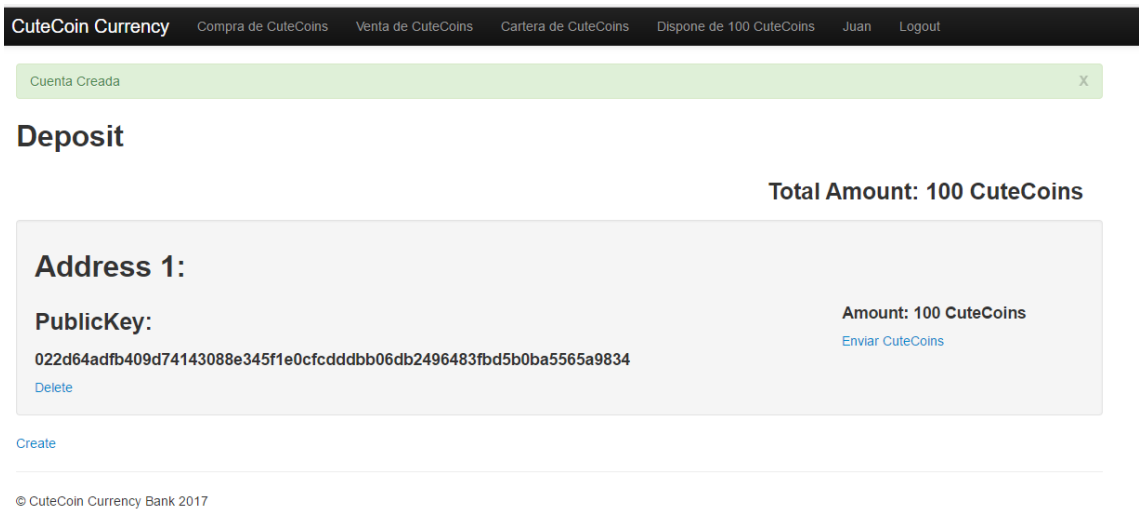
Como podemos ver, actualmente no disponemos de ninguna moneda virtual.



Vamos a nuestra cartera a crear una dirección donde guardar nuestras monedas virtuales.



Creamos una cuenta.



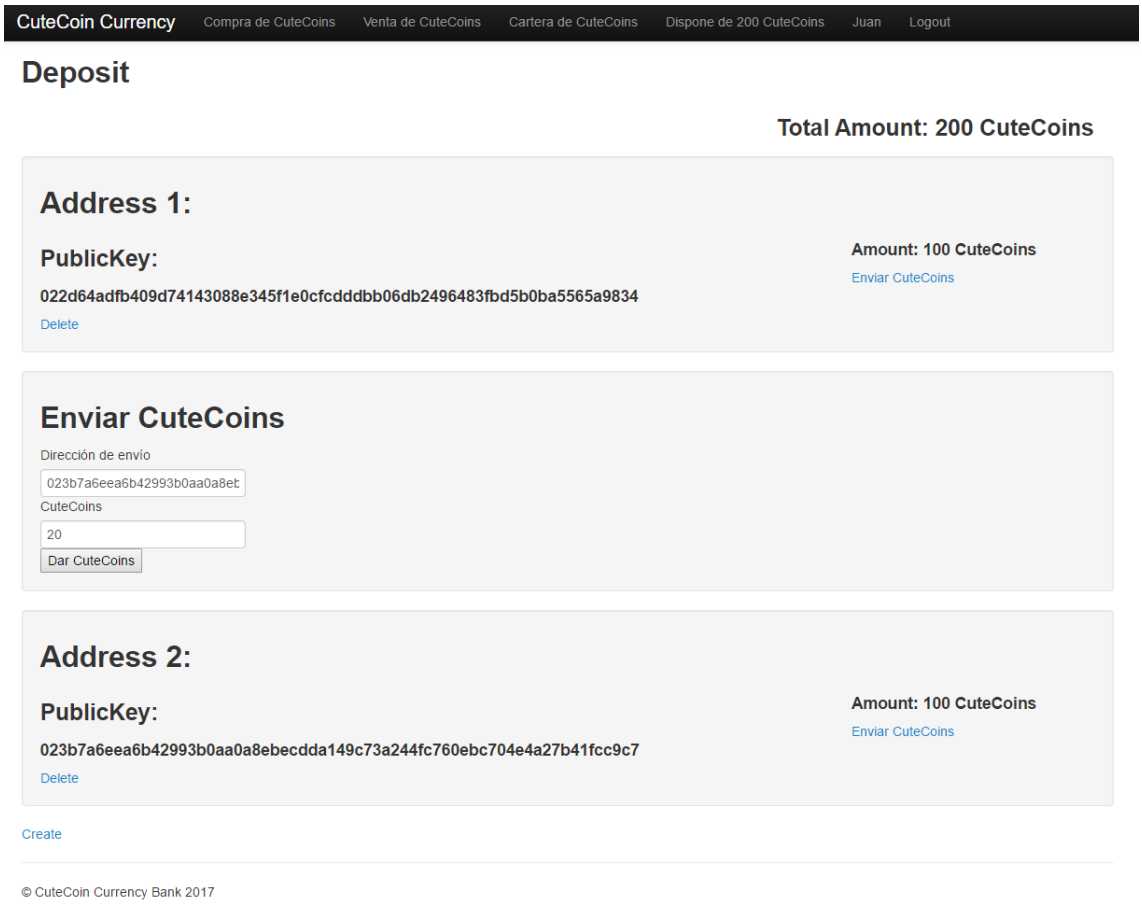
Como podemos ver, nos ha generado una cuenta con una dirección pública (la cual debemos de utilizar para nuestras transacciones). También por defecto, creamos la cuenta con 100 monedas virtuales debido a que en nuestra simulación los mineros no reciben monedas por la minería que realizan.

Internamente, en nuestra base de datos guardamos además de la dirección pública la clave privada.

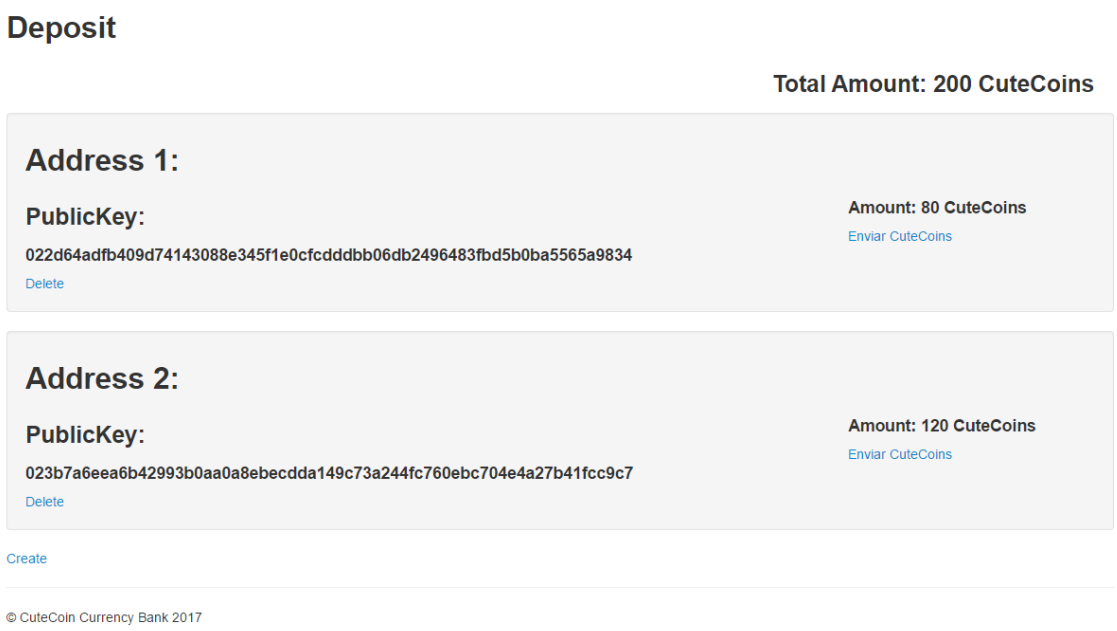
	ID	FKWALLET	ADDRESSKEY	CRYPTOSECRET	AMOUNT
1	1	0	0382b145daf1b9759e72ecc6906d120495b0e257329509c2...	cRALc3n4RqP6WkCxdH1Honxuremm8X6Hohtdc6wTL343qXCgPjc1	493
2	2	0	02b0842f9b3d64f4dabf62fdbaf1ec12b31981cc17bfd09fd5c...	cTD1VsZoEUtTJG7NQYCa7vH9oFo6tMQaU3UvRszi4J714xD6jc7c	7
3	3	1	022d64adfb409d74143088e345f1e0cfcdddbb06db2496483f...	cT4o8Ptwq4pXVr22diH83ZFdaovuh2JYwRxWHVHt3KWPWWH7996X	100

Si nuestra cartera fuera una cartera local, necesitaríamos también dar al usuario la información sobre su clave privada pero al tratarse de una web, por lo general no se suele dar la información de la clave privada (incluso en las webs de bancos de Bitcoin, gestionan internamente las claves privadas).

Creamos otra dirección para guardar monedas virtuales y vamos a hacer un traspaso de 20 monedas de una cuenta a otra.

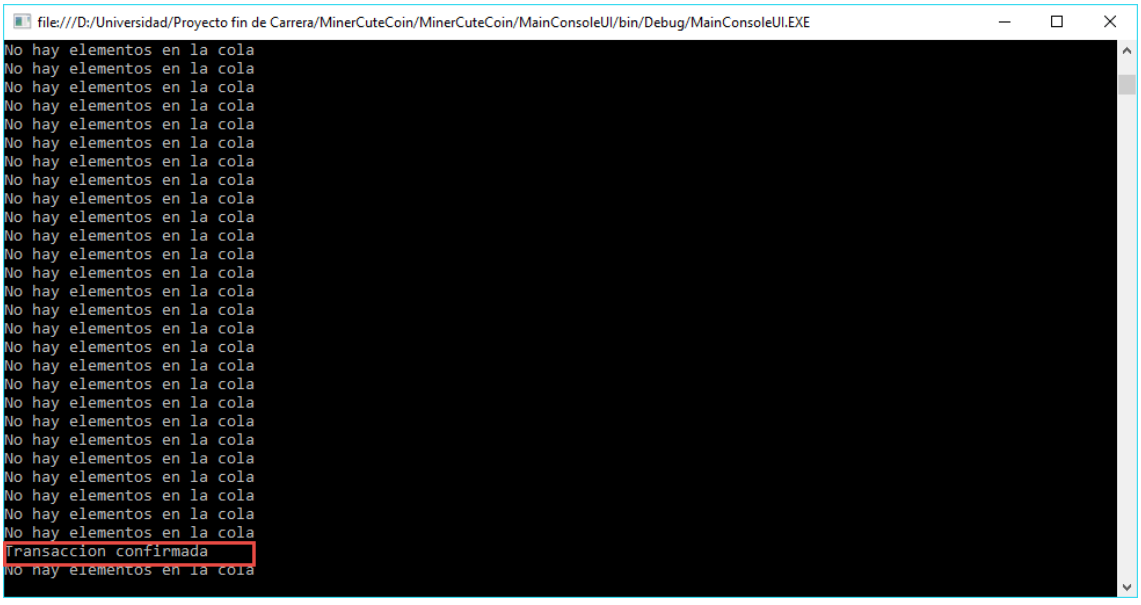


Los enviamos y comprobamos que se han recibido correctamente.



Veamos lo que sucede internamente en nuestro sistema.

El usuario realiza una transacción a través de nuestro banco, el cual va asociado a Blockchain. Nuestra red añade en la cola que hay una transacción pendiente de confirmar, por lo que nuestro minero obtiene la transferencia y la confirma.



A su vez, cuando la transacción se confirma se genera un nuevo bloque dentro de Blockchain con la transacción con el nuevo Hash y el hash del anterior bloque (al igual que sucede con Bitcoin).

	ID	HASH	HASHANTBLOCK
1	0	1Init234	0
2	1	8836770	1Init234
3	2	22379119	8836770

Dentro de Blockchain guardamos también la información sobre la transacción realizada. Junto con las direcciones públicas que han hecho la transacción, la cantidad, el bloque asociado y el Hash propio de la transacción (recordemos que tanto los bloques como las transacciones tienen su Hash propio y único).

	ID	FKBLOCK	HASH	PUBKEYPAY	PUBKEYRECEIBER	AMOUNT
1	0	0	1234	1234	4321	12
2	1	1	85292e4f7a20d78ec556a6d14a739c48cce50c5b	0382b145daf1b9759e72ecc6906d120495b0e257329509c2...	02b0842f9b3d64f4dabf62fdbaf1ec12b31981cc17bfd09fd5c...	2
3	2	2	a3da450115bea8b6047f4c787ebdd7357cdbef7e	022d64adfb409d74143088e345f1e0cfcdddbb06db2496483f...	023b7a6eea6b42993b0aa0a8ebecdda149c73a244fc760eb...	20

Esto es lo que sucede dentro de la base de datos de Blockchain, ¿qué sucede en la base de datos del proyecto del Banco?

	ID	FKWALLET	ADDRESSKEY	CRYPTOSECRET	AMOUNT
1	1	0	0382b145daf1b9759e72ecc6906d120495b0e257329509c21e329a129c...	cRALc3n4RqP6WkCxdH1Honxuremm8X6Hohtdc6wTL343qXCgPjc1	493
2	2	0	02b0842f9b3d64f4dabf62fdbaf1ec12b31981cc17bfd09fd5c60664bd636...	cTD1VsZoEUtTJG7NQYCa7vH9oFo6tMQaU3UvRszi4J714xD6jc7c	7
3	3	1	022d64adfb409d74143088e345f1e0cfcdddbb06db2496483fbd5b0ba556...	cT4o8Ptwq4pXVr22diH83ZFdaovuh2JYwRxWHVHt3KWPWWH7996X	80
4	4	1	023b7a6eea6b42993b0aa0a8ebecdda149c73a244fc760ebc704e4a27b...	cVjLyBG4zND3rG7RFRtaaPWN6j32NdfQJm7uCuSew8g9SKQekv9d	120

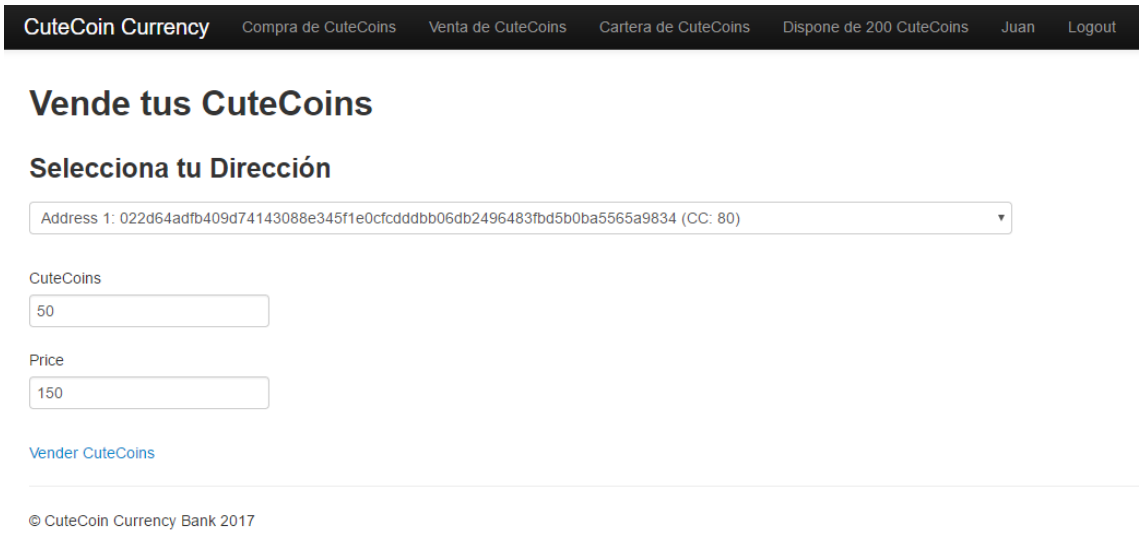
Como podemos ver, ahora las cuentas asociadas a este usuario disponen de 80 y 120 monedas respectivamente.

Y la información de la transacción realizada.

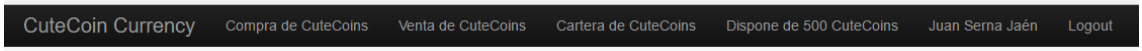
	ID	FKADDRESS	TYPE	HASH	AMOUNT	PRICE
1	0	1	0	d21633ba23f701181...	2	0
2	1	2	1	d21633ba23f701181...	2	0
3	2	1	0	d21633ba23f701181...	1	0
4	3	2	1	d21633ba23f701181...	1	0
5	4	1	0	d21633ba23f701181...	1	0
6	5	2	1	d21633ba23f701181...	1	0
7	6	1	0	d21633ba23f701181...	1	0
8	7	2	1	d21633ba23f701181...	1	0
9	8	1	0	d21633ba23f701181...	2	0
10	9	2	1	d21633ba23f701181...	2	0
11	10	3	0	d21633ba23f701181...	20	0
12	11	4	1	d21633ba23f701181...	20	0

En la base de datos se generan dos valores debido a que obtiene la información tanto del que transfiere las monedas como del que las recibe (podemos observar que el “TYPE” varía de una a otra).

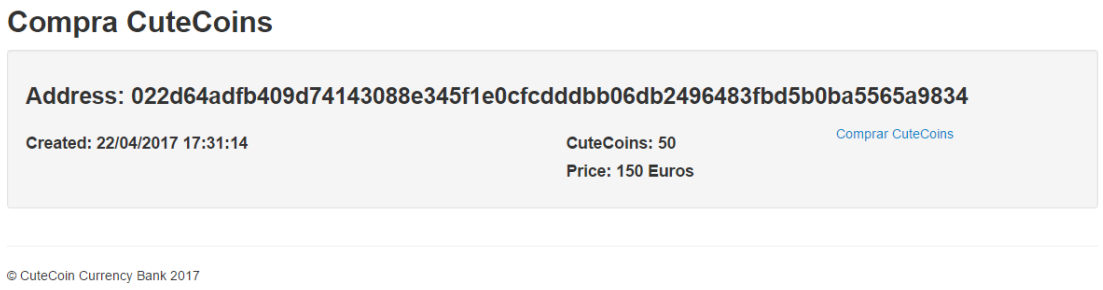
Ahora vamos a hacer una venta de algunas de nuestras monedas virtuales. Vamos a la pestaña de Venta y seleccionamos nuestra dirección (actualmente tenemos dos), la cantidad de monedas puestas a la venta y el precio por el cual queremos venderlas.



Me conecto con otro usuario.

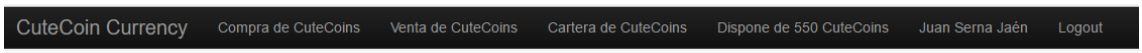


Y voy al apartado de Compra.

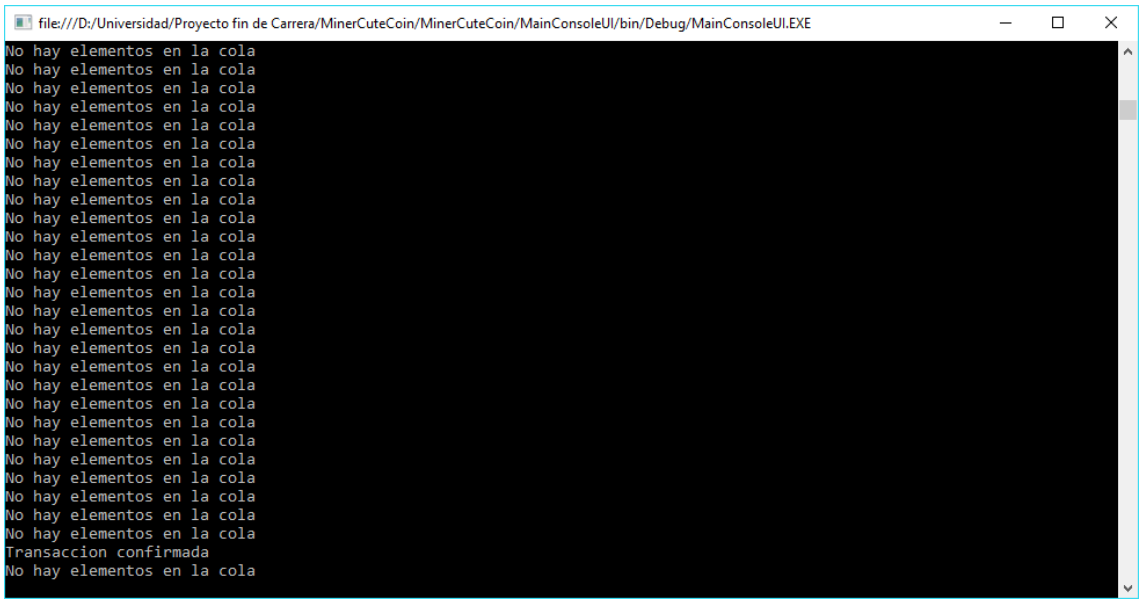


Veo que una dirección está vendiendo monedas virtuales, le doy click a comprar y selecciono la dirección de mi cartera en la que quiero añadir las monedas.

Puedo ver que ahora mismo dispongo de 50 monedas virtuales más.



Y que el minero ha encontrado una nueva transacción que añadir a la cadena de bloques.



Comprobamos en la base de datos de Blockchain cómo se ha generado un nuevo bloque.

	ID	HASH	HASHANTBLOCK
1	0	1Init234	0
2	1	8836770	1Init234
3	2	22379119	8836770
4	3	59192434	22379119

Y vemos la transacción que lleva dicho bloque.

	ID	FKBLOCK	HASH	PUBKEYPAY	PUBKEYRECEIBER	AMOUNT
1	0	0	1234	1234	4321	12
2	1	1	85292e4f7a20d78ec556a6d14a739c48cce50c5b	0382b145daf1b9759e72ecc6906d120495b0e257329509c2...	02b0842f9b3d64f4dabf62fdbaf1ec12b31981cc17bfd09fd5c...	2
3	2	2	a3da450115bea8b6047f4c787ebdd7357cdbef7e	022d64adfb409d74143088e345f1e0cfcdddbb06db2496483f...	023b7a6eea6b42993b0aa0a8ebecdda149c73a244fc760eb...	20
4	3	3	436fbf2f842c47602f14a27739db6b7092c25721	022d64adfb409d74143088e345f1e0cfcdddbb06db2496483f...	0382b145daf1b9759e72ecc6906d120495b0e257329509c2...	50

Por lo que vemos, el bloque se ha generado correctamente. Veamos finalmente en la base de datos de nuestro banco que la información de la transferencia también es la correcta.

	ID	FKADDRESS	TYPE	HASH	AMOUNT	PRICE
1	0	1	0	d21633ba23f70118...	2	0
2	1	2	1	d21633ba23f70118...	2	0
3	2	1	0	d21633ba23f70118...	1	0
4	3	2	1	d21633ba23f70118...	1	0
5	4	1	0	d21633ba23f70118...	1	0
6	5	2	1	d21633ba23f70118...	1	0
7	6	1	0	d21633ba23f70118...	1	0
8	7	2	1	d21633ba23f70118...	1	0
9	8	1	0	d21633ba23f70118...	2	0
10	9	2	1	d21633ba23f70118...	2	0
11	10	3	0	d21633ba23f70118...	20	0
12	11	4	1	d21633ba23f70118...	20	0
13	12	3	0	d21633ba23f70118...	50	150
14	13	1	1	d21633ba23f70118...	50	150

Y con esto concluyen nuestros ejemplos de uso, como podemos ver en este proyecto de simulación solo se ha utilizado un minero y éste no se lleva ninguna recompensa a cambio de descifrar y crear el bloque pero podemos hacernos una idea bastante acertada de cómo funciona una moneda virtual en la vida real.

7. Conclusiones

El proyecto de simulación de una moneda virtual puede ser una primera fase para que la universidad de alicante tenga una base para dar a conocer cómo funciona una moneda virtual y, lo que es más importante; cómo funciona Blockchain.

Los conocimientos aplicados para esta aplicación pueden ser de utilidad por ejemplo para un sistema de "tokens" con el que recompensar a alumnos por su esfuerzo y dedicación en las diferentes asignaturas.

También se podría utilizar en asignaturas de redes para explicar el concepto de red distribuida y criptografía. Ver cómo se comporta la red a cambios o a conexión / desconexión de nuevos dispositivos.

Y uno de los conceptos más interesantes sería el de crear una red para compartir información no ya a través de la propia universidad de Alicante, si no crear una red a nivel nacional (incluso internacional) que perdure con el tiempo y sea segura. Se podría crear una red de alumnos que pueda ser utilizada en cualquier universidad, aportando información sobre su perfil, lo que le interesaría investigar, sus cualidades, …

En definitiva, la realización de esta práctica me ha enseñado lo útil que puede llegar a ser una red de usuarios conectada a nivel internacional y cómo se puede utilizar el manejo de la información para construir procesos útiles para la sociedad.

8. Bibliografía y referencias

Unimooc. ¿Qué son los Bitcoins? - *http://unimooc.com/monedas-virtuales-que-es-bitcoin/*

QueesBitcoin. Bitcoin, la moneda que está cambiando el mundo - *https://www.queesbitcoin.info/*

Wikipedia. Bitcoin - *https://es.wikipedia.org/wiki/Bitcoin*

AprenderInternet. ¿Qué es Bitcoin? - *http://aprenderinternet.about.com/od/ECommerce/a/Que-Es-Bitcoin.htm*

MapofCoins. Bitcoin Map - *http://mapofcoins.com/bitcoin*

CoinTelegraph. History of Cryptocurrency - *http://cointelegraph.com/news/history-of-cryptocurrency-from-bitcoins-inception-to-the-crypto-boom*

Wikipedia. Criptografía asimétrica - *https://es.wikipedia.org/wiki/Criptograf%C3%ADa_asim%C3%A9trica*

Bitcoin. ¿Quién creó Bitcoin? - *https://bitcoin.org/es/faq#quien-creo-bitcoin*

P2PFoundationWiki. Blockchain - *http://wiki.p2pfoundation.net/Blockchain*

Bit2Me. ¿Qué es la Cadena de Bloques (Blockchain)? - *http://blog.bit2me.com/es/que-es-cadena-de-bloques-blockchain/*

Wikipedia. Secure Hash Algorithm - *https://es.wikipedia.org/wiki/Secure_Hash_Algorithm*

Bitcoin. ¿Cómo funciona? - *https://bitcoin.org/es/como-funciona*

Blockchain. Blockchain - *https://blockchain.info/*

BestBitcoinExchange. Sitios para intercambiar bitcoin - *https://www.bestbitcoinexchange.io/*

Humaniq. Humaniq - *https://humaniq.io*

Everledger. Everledger - *https://www.everledger.io/*

ElDiario. Qué es el blockchain y por qué los bancos invierten millones en esta tecnología - *http://www.eldiario.es/economia/blockchain-bancos-quieren_0_572893447.html*

ElEconomista. Santander y otros grandes bancos lanzan su bitcoin utilizando la tecnología blockchain - *http://www.eleconomista.es/empresas-finanzas/noticias/7782103/08/16/Economia-Santander-se-une-a-cinco-entidades-para-promover-el-uso-de-dinero-digital-entre-entidades-financieras.html*

MicrosoftAzure. Soluciones Blockchain - *https://azure.microsoft.com/en-us/solutions/blockchain/*

Hyperledger. Hyperledger - *https://www.hyperledger.org/*

Ethereum. Ethereum - *https://www.ethereum.org/*

OroYFinanzas. ¿Qué es la tecnología blockchain de Ethereum? - *https://www.oroyfinanzas.com/2016/04/que-blockchain-ethereum/*

DiarioBitcoin. Lista actualizada de los usos de ethereum - *http://www.diariobitcoin.com/index.php/2016/05/25/una-lista-actualizada-de-los-usos-de-ethereum/*

Augur. Augur - *https://augur.net/*

Akasha. Akasha - *https://akasha.world/*

Lo3Energy. Projects - *http://lo3energy.com/projects/*

MsdnMicrosoft. Entity Framework - *https://msdn.microsoft.com/es-es/library/bb399567(v=vs.110).aspx*